CONTROL OF REAL-TIME MULTIMEDIA APPLICATIONS IN BEST-EFFORT NETWORKS

A Dissertation

by

DAN YE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2006

Major Subject: Mechanical Engineering

CONTROL OF REAL-TIME MULTIMEDIA APPLICATIONS IN BEST-EFFORT NETWORKS

A Dissertation

by

DAN YE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	A. Parlos
Committee Members,	S. Jayasuriya
	W.-J. Kim
	E. Sanchez-Sinencio
Head of Department,	D. O'Neal

December 2006

Major Subject: Mechanical Engineering

ABSTRACT

Control of Real-time Multimedia Applications
in Best-effort Networks. (December 2006)
Dan Ye, B.S., Tsinghua University;
M.S., Tsinghua University
Chair of Advisory Committee: Dr. A. Parlos

The increasing demand for real-time multimedia applications and the lack of quality of service (QoS) support in public best-effort or Internet Protocol (IP) networks has prompted many researchers to propose improvements on the QoS of such networks. This research aims to improve the QoS of real-time multimedia applications in public best-effort networks, without modifying the core network infrastructure or the existing codecs of the original media applications.

A source buffering control is studied based on a fluid model developed for a single flow transported over a best-effort network while allowing for flow reversal. It is shown that this control is effective for QoS improvement only when there is sufficient flow reversal or packet reordering in the network.

An alternate control strategy based on predictive multi-path switching is studied where only two paths are considered as alternate options. Initially, an emulation study is performed, exploring the impact of path loss rate and traffic delay signal frequency content on the proposed control. The study reveals that this control strategy provides the best QoS improvement when the average comprehensive loss rates of the two paths involved are between 5% and 15%, and when the delay signal frequency content is around 0.5 Hz. Linear and nonlinear predictors are developed using actual network data for use in predictive multi-path switching control. The control results show that predictive path switching is better than no path switching, yet no one predictor

developed is best for all cases studied. A voting based control strategy is proposed to overcome this problem. The results show that the voting based control strategy results in better performance for all cases studied. An actual voice quality test is performed, proving that predictive path switching is better than no path switching.

Despite the improvements obtained, predictive path switching control has some scalability problems and other shortcomings that require further investigation. If there are more paths available to choose from, the increasing overhead in probing traffic might become unacceptable. Further, if most of the VoIP flows on the Internet use this control strategy, then the conclusions of this research might be different, requiring modifications to the proposed approach. Further studies on these problems are needed.

To My Parents

ACKNOWLEDGMENTS

Thank Dr. Parlos for his advises during my Ph.D. study. Thank Dr. Jayasuriya, Dr. Kim, and Dr. Sanchez-Sinencio for their help during my research. Thank my friends Lin, Aninda, Ram, Ben, and Choi for their help all these years. Thank all the graduated former student from this group, Srika, Tolis, Vivek, Deepanker, Rajash, Mallik, and Yash, for their pioneer work related to my research. Thank all my friends and GSers for their friendship and bring happiness into my daily life. Thank my parents for supporting me on everything all these years. Thank Cloudia for her spiritual companionship. And thank Phoero for showing me the merits of life.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

TABLE Page

TABLE Page

TABLE Page

TABLE Page

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivation

The past decade has seen an increasing number of real-time multimedia applications running on the Internet. Because of the cost efficiency and scalability of best-effort networks, more and more companies and individuals are using VoIP services deployed over the Internet to replace traditional toll phones. The current revenue from VoIP services is around US$3 billion, and is expected to reach US$18 billion by 2010 [1]. The current total market for VoIP equipment is around US$4.8 billion. It is expected to reach US$5.5 billion by 2007 and fall back to US$3 billion towards the end of this decade [2].

Comparing with the toll phone, VoIP costs much less, because it is sharing the same networks with other data transmission applications. But best-effort networks have no guarantees on packet delay, loss rate, and jitter [3]. When there is congestion, the perceptive quality of VoIP is hampered. This has prohibited a lot of potential users from switching from toll phone to VoIP. The problem of improving the Quality of Service (QoS) for VoIP applications in public best-effort networks has received a lot of research interest.

Research in this area includes router based solutions, end-to-end solutions, and overlay network solutions [4]. The router based solutions require access to the information inside the networks. They are more likely to provide better improvements for VoIP QoS over the Internet. Their shortcoming is that their implementation normally requires replacement of most of the existing routers. Such replacement is very

---

The journal model is *IEEE Transactions on Automatic Control*.

unlikely to be carried out within the next few years. The end-to-end solutions do not require replacement of any routers and are more likely to be implemented. Their shortcoming is that accessibility to the information inside the networks is poor at the end-user side. The improvements that can be achieved using end-to-end solutions are likely to be quite limited. The overlay network solutions [5] behave like the router based solutions on the nodes where they are implemented. Between a pair of nodes other than the ones they are implemented they behave like the end-to-end solutions. They are likely to have better performance than the end-to-end ones. And they are more likely to be implemented than the router based ones.

If there are multiple-path connections on a given network, then there is a *path-diversity* available. It is possible to improve the QoS by dynamically switching among available paths, an approach called *path switching*. Recently, Tao [6] has done some work with this method.

B. Problem Definition

Problem Definition: *The objective of this research is to develop a data driven controller that can be implemented independent of encoding scheme in the end-to-end system to improve the QoS of best-effort networks for real-time multimedia application.*

The control approach developed in this research should be able to work with any currently available real-time multimedia application, regardless of the codecs being used. Thus, it should be implementable as middleware. This control approach should also be an end-to-end solution, without changing the core infrastructure of the networks, in order to be scalable, rapidly deployable, and widely implementable. Because reactive controls suffer from the round trip time (RTT) delay of networks,

this control approach should be predictive as well.

Two possible solutions have been studied in this research to achieve this objective:

1. Source buffering based predictive control: In this solution, the packets generated from an application are held in a source side buffer whose send rate is determined by the predictive controller. The controller adapts its send rate according to the predicted network conditions.

2. Path switching: In this solution, the path-diversity available at the source is exploited. The controller predicts the network condition of the available paths and attempts to send the application packets through the best available path.

C. Research Objectives

The main objective of this research is to find active control means to improve the QoS of real-time multimedia applications transported over public best-effort networks. The specific objectives are to:

- Derive a fluid model for a single flow transported over a best-effort network while allowing for flow reversal.

- Investigate the possibility of improving the QoS of real-time multimedia applications through predictive source buffering.

- Study the possibility of improving the QoS of real-time multimedia applications through predictive path switching.

- Investigate the factors that impact the theoretically achievable improvement by predictive path switching.

- Develop predictors based on dynamic system models for predicting various information signals from the measurements of the network paths.
- Develop predictive path switching controllers to improve the QoS of real-time multimedia applications transported over public best-effort networks.

D. Contributions

The contributions of this dissertation are the following:

1. Single path control: Under certain assumptions, a continuous fluid model of a single flow transported over a best-effort network is developed which allows for flow reversals. This dissertation proves that source buffering based predictive control is effective for improving the QoS of real-time multimedia applications only when there is sufficient flow reversal in the network.
2. Multipath switching control: A voting based predictive path switching control is developed to improve the QoS of real-time multimedia applications. This dissertation demonstrates that predictive path switching control can improve the QoS in a meaningful way in actual networks.

E. Dissertation Outline

Chapter II gives a brief literature review of previous research. Section A covers the current state of best-effort networks. Section B covers some of the current research on improving the QoS of these networks. Section C reviews methods used for improving the QoS in multimedia applications. Section D covers the research targeting improvements in QoS of VoIP applications. Section E reviews the resources and research on modeling the Internet. Section F reviews some of the methods developed

for controlling systems with time-varying time-delay.

Chapter III summarizes the available speech quality evaluation methods. Section A reviews both the subjective and objective speech quality evaluation methods. The formulae for mapping the objective scores to the subjective scores are given. Section B discusses the problem of estimating the E-model parameters for the Speex codec used in this study.

Chapter IV investigates the possibility of improving the QoS in real-time multimedia applications using a fluid model based single flow control. The assumptions behind this approach are given in Section A. The fluid model of a single flow transported over best-effort networks with no flow reversal is presented in Section B. The fluid model of a single flow with flow reversal is derived in Section C. The discrete versions of the fluid models are derived in Section D for different assumptions of the form of the input flow. The source buffering based predictive controller is investigated in Section E. A detailed literature review of flow reversal on the Internet is presented in Section F. Section G discusses the impact of losses in the network on the source buffering control.

Chapter V presents the predictive path switching control method. It gives the problem statement, proposes the use of a dynamic system model for prediction, gives the general assumption behind this method, introduces several concepts used in this method, and conducts an emulation study of both the impact of path comprehensive loss rate and traffic delay signal frequency content on predictive path switching control. A review of previous research in this area and the problem statement are given in Section A. Section B gives the general assumption behind this method and several concepts used in this method. Section C discusses the method used for the emulation study of the impact of path comprehensive loss rate and traffic delay signal frequency content on this method. Section D shows the results of the emulation study

and discusses some observations from them.

Chapter VI discusses the problems arising in the data collection process for this study from actual networks. A preliminarily investigation on the possibility of improving the VoIP QoS through path switching with the data collected from actual networks is conducted. The problems regarding actual network data collection and their solutions are presented in Section A. Section B gives a preliminary investigation of the possibility of improving the VoIP QoS through path switching using the actual network data collected.

Chapter VII discusses the problems related to the predictor development and presents the development results. Section A discusses the requirements of the predictors developed for predictive path switching control. Section B investigates the relation between the prediction/switching interval and the control results. Section C investigates the development of predictors for the predictive path switching control problem. The information signals used in the prediction, the different types of predictors, and the parameter selection problem are discussed. Finally the prediction results are presented.

Chapter VIII studies the performance of predictive path switching control with the developed predictors. Section A presents the control logic of the controllers. Section B gives the control results. Section C proposes and investigates a voting based predictive control method. Section D discusses the problem of the predictor performance evaluation criteria.

Chapter IX conducts an actual voice quality test using the voting based predictive path switching control. Section A gives a description of the voice quality test system. Section B presents the test results.

Chapter X studies some implementation aspects of the predictive path switching control method. Section A investigates the possibility of using a set of unified

predictors for all paths in the predictive path switching control. Section B discusses the trade-off between probing rate, prediction complexity, and resulting voice quality. Section C discusses the switching control problem when there are multiple paths available in the networks.

Chapter XI gives the summary and conclusions of this research. It presents the contributions and limitations of this work. Finally it gives some suggestions for future work.

CHAPTER II

LITERATURE REVIEW

A. Current Best-effort Networks

Both local-area and wide-area network traffic shows self-similarity [7,8]. The Internet which is a best-effort packet-switched network also has a self-similar traffic delay profile [9]. Network Calculus [10], which is a collection of results based on Min-Plus algebra, has been developed to analyze networks in a deterministic framework. It can be used to determine tight bounds on delay and backlog.

Transmission Control Protocol (TCP) is a reliable transmission protocol used by the majority of current data traffics transported over the Internet. A congestion control algorithm, Additive Increase Multiplicative Decrease (AIMD), was added by Van Jacobson in TCP Tahoe. Fast recovery is implemented in TCP Reno. Selective Acknowledgment has been proposed as TCP SACK. "Partial acknowledgments" has been implemented in TCP NewReno. There are several survey papers available on TCP congestion control [11, 12]. Chiu and Jain [13] proved that AIMD converges to an efficient and fair state regardless of the starting point of the network. Bansal and Balakrishnan [14] proposed a nonlinear version of a congestion control algorithm for steaming audio and video applications.

In order to achieve congestion control without dropping a packet, Explicit Congestion Notification (ECN), delay-based congestion avoidance, and accumulation-based congestion avoidance algorithms [15] have been proposed. One famous delay-based TCP is TCP Vegas. But Martin et al. [16] claimed that a single deployment of Delay-based Congestion Avoidance (DCA) algorithm is not a viable enhancement to TCP. This might be because a reactive DCA can not respond to network condition

changes fast enough. Xia et al. [17] suggested that TCP Vegas can also be viewed as an Accumulation-based Congestion Control (ACC).

Packet reordering is considered as packet loss by current TCP flavors and leads to poor performance. Bennett and Partridge showed that with the parallelism in the Internet paths, packet reordering is a natural network behavior [18]. The report from Internet End-to-end Performance Monitoring (IEPM) group supported their observation that there is significant amount of packet reordering in the Internet [19]. However, Jaiswal et al. [20] and Gharai et al. [21] reported much lower rate of packet reordering in the Internet compared to the previous reports. One possibility is that Jaiswal and Gharai were using larger inter-departure time. Both Gharai et al. [21] and Bellardo and Savage [22] observed that inter-departure time less than tens of microseconds tends to lead to more packet reordering. As the packet reordering problem is important for TCP in high speed networks, researchers are trying to provide metrics for packet reordering [23, 24]. Recently the Internet Engineering Task Force (IETF) has proposed a new standard for packet reordering metrics [25]. Bohacek et al. proposed another TCP flavor to deal with packet reordering [26].

User Datagram Protocol (UDP) is the protocol being used by most multimedia applications. The increasing use of UDP flows, which has no congestion control, may cause unfairness to TCP flows and may even cause congestion collapse [27]. TCP-friendly, i.e. matching the throughput with that of TCP traffic under the same condition, has been proposed. Studies on TCP indicate that its throughput is inversely proportional to the product of Round Trip Time (RTT) and square root of loss event rate [28]. The resulting new protocol is TCP Friendly Rate Control (TFRC). A survey of approaches on TCP-friendliness can be found in [29].

Asynchronous Transfer Mode (ATM) is a cell relay network protocol which encodes data traffic into small fixed-sized cells instead of variable sized packets as in

packet-switched networks, such as the Internet Protocol (IP). Available Bit Rate (ABR) service has enabled ATM networks to transfer the normal IP traffic as best-effort packet-switched networks. Kolarov and Ramamurthy [30] designed a feedback control system to support ABR service for a time-invariant system with constant delays. A time-varying linear feedback system model for the same problem was studied by Sichitiu et al. [31]. Mascolo [32] tried to maximize the utilization of an ATM network link and avoid congestion by regulating the bottleneck queue level, and used a Smith Predictor to overcome the delays. Gu et al. [33] attacked this problem with a model-based predictive controller.

Active Queue Management (AQM) tries to indirectly keep output flow rate in packet-switched networks close to the full link capacity while avoiding congestion by regulating bottleneck router queue level. A well known AQM policy is Random Early Detection (RED). Misra et al. [34] developed a fluid model for the interactions of a set of TCP flows and AQM routers. Hollot et al. [35] linearized this model at steady state, and analyzed the linear model with classical control design methods. Quet et al. [36] built a rate-based model, which took the uncertain time-varying time-delays in the channels into account, and built an H_{inf} robust controller based on that model. Aweya et al. [37] has used multi-step neural networks to predict the bottleneck queue level and control the source rate over a finite prediction horizon.

B. Quality of Service in Best-effort Networks

Delay, which is the time taken for packets to move from the sender to the receiver, can impact interactivity. Jitter, which is the random variation in the end-to-end delay, can cause gaps in the playout of an audio stream, or result in a choppy appearance in a video display. Packet loss, which refer to the packets that do not arrive from

the sender to the receiver, can significantly degrade the presentation quality of both video and audio. Thus to guarantee the Quality of Service (QoS) is to control these three factors, delay, jitter and packet loss [4].

One simple and effective way to provide QoS is over-provisioning. In fact, one of the factors that has delayed the demand for QoS control has been the transition of the Internet infrastructure from heavy congestion to over-provision in the backbones in the late 1990s [38].

The current research on QoS control can be classified into network-based techniques and end-system based techniques [4]. In the class of network-based techniques, some are based on new IP architectures, such as Integrated services (IntServ), Differentiated services (DiffServ), and Active Networks (AN); some are dependent on other new network infrastructures including IP over ATM and IP over Frame Relay; and some belong to new router/routing technology such as IP switching and MultiProtocol Label Switching (MPLS) [4].

The IntServ approach provides QoS through per-flow end-to-end resource reservation. The resource reservation is done through the Resource Reservation Protocol (RSVP). Schedulers are used to meet the delay deadlines. IntServ provides the best guarantee for QoS. But it has to keep states for each flow, which is a heavy burden. Also the reservation may result in insufficient availability of peak resources [4].

In DiffServ approach, traffic is classified into several classes according to their demand of QoS, and are provided with different levels of service accordingly. There will be improvement in the QoS for classes with higher priority, but an end-to-end delay bound as in IntServ cannot be assured [4].

In the AN approach, more complexity is put into core routers. The AN nodes are programmable. They will go up even to the application-layer of packets to provide QoS [4].

End/Edge-based techniques include Forward Error Correction (FEC), packetization, error concealment, rate control, and layered coding [4]. FEC is an open-loop passive loss recovery method used at the sender side. Parity coding or Reed-Solomon coding are added to packets, so that when a packet is lost, it can still be recovered at the receiver side. Different packetization methods combined with FEC can result in various combination of redundancy and robustness [39]. The effect of delay and packet loss on VoIP in the presence of FEC has been investigated in detail in the literature [40]. But FEC itself consumes bandwidth. There is a trade-off between the robustness against losses with multimedia quality [41].

Error concealment schemes produce a replacement for a lost packet at the receiver side. This can be done either by inserting silence, noise, repeating packet, or by interpolating between the received packets [42].

Bandwidth adaptation or adaptive rate control is to adapt the application send rate to the available bandwidth of the networks. This can be done by adapting the media quality with the bandwidth [43], through a selection of the optimal combination of compression and FEC [44], or through a format transcoding and media re-synchronization [45].

Layered coding is a receiver-driven adaptation, proposed mainly for multicast applications. The media is compressed into one base layer and several enhancement layers. The receivers choose how many layers to subscribe to get the best quality within its capacity [4].

The increasing number of Internet Service Providers (ISPs) has made it possible for a single source to have multiple Internet connections from different ISPs. An ad-hoc wireless network is also capable of having multiple paths. One way to utilize multiple paths is to send duplicate packets through multiple paths to reduce losses [46, 47] or reduce delay jitter [48]. Another way to utilize them is through path switching.

Savage et al. [49] reported that the currently used routing methods are not the best. In $30\% \sim 80\%$ of the cases, there is a better alternative path. Tao et al. [50] showed that path switching can help in loss rate reduction. They demonstrated it with a streaming video application [51].

In industry it is common for servers to have multiple connections to the Internet and then use a route optimization product to choose the best connection to improve their performance [52]. Kang and Nath [53] have proposed QoS control for voice traffic by toggling it between circuit and packet cellular networks. Skype [54], a peer-to-peer VoIP product, is able to keep multiple connection paths open and dynamically choose the best one. In the literature there is a emerging interest to implement QoS control techniques as middleware architectures between the network layer and the application layer [55].

Besides the network based and end/edge based methods, an interesting overlay network based method has been proposed [5, 56]. The location of the Internet bottlenecks can be classified into first mile, backbone, and last mile [57]. The overlay network was originally proposed for the backbone bottleneck problems. Yet it has become a good way of combining the network based methods and end/edge based methods. Subramanian et al. [58] combined forward error correction (FEC) and automatic repeat request (ARQ) with an overlay structure to come-up with the OverQoS architecture for enhancing the Internet QoS. They also implemented DiffServ in the overlay nodes [59]. Li and Mohapatra [60] used an overlay structure for QoS-aware routing. They reported that the overlay topology has significant impact on the routing performance [61]. Ma et al. [62] used an overlay structure for video streaming, and Gu et al. [63] used an overlay structure for multicast applications.

C. Multimedia Applications

In streaming applications there are six basic blocks: media compression, application layer QoS control, continuous media distribution services, streaming servers, media synchronization mechanisms, and protocols for streaming media [64]. One way to provide the QoS is to adapt the send rate with the available bandwidth. Jain and Dovrolis [65] measured the available bandwidth by increasing the stream's rate to the point the one-way delay goes up. Wu et al. [66] used RTP/RTCP feedback to estimate the viable bandwidth. Sony's video conferencing system has used rate adaption with active repeat request [67].

Sun et al. [68] argued that adaptive rate control may not react fast enough. They proposed a predictive rate control method, which transfers more data ahead of congestion to ensure there are sufficient packets at the receiver during congestion . Wee et al. [69] tried to reduce the number of late frames for video streaming by transmitting I and P frames first.

A sender side buffering was used by Liew and Tse [70] to transfer variable bit rate (VBR) video over a constant bit rate (CBR) channel. Its image quality was adjusted according to the buffer level. A receiver side buffering was used by Chakrabarti and Wang [71]. In their method, the receiver sends the compression parameter back to the sender to regulate the receiver buffer level and reduce the jitter.

D. VoIP Control

VoIP is a good example of an interactive multimedia application. There are two important VoIP call signaling protocols: ITU-T Rec. H.323 [72] and Session Initiation Protocol (SIP) [73]. H.323 is a set of protocols for voice, video and data conferencing over packet-based network. SIP is an application-layer control signaling protocol for

creating, modifying, and terminating sessions with one or more participants.

Voice quality can be affected by the choice of codec, echo control, packet loss, delay, jitter and design of network [3]. The quality of a voice call is often measured by subjective testing under controlled conditions using a large number of listeners to determine a Mean Opinion Score (MOS) [74]. Objective test methods have also been developed, including signal-based methods, such as Perceptual Speech Quality Measure (PSQM, ITU-T P.181), Measuring Normalizing Blocks (MNB), Perceptual Analysis Measurement System (PAMS), Perceptual Evaluation of Speech Quality (PESQ, ITU-T P.182), and parameter-based methods, such as the E-model [75].

PESQ, which has replaced PSQM as the new standard, has some time delay identification techniques to deal with delay variations [76]. Shim et al. [77] studied how do delay and packet loss impact voice quality in VoIP using the PESQ score. Furuya et al. [78] used PESQ to study the relation between VoIP quality and the bandwidth of the bottleneck link, the size of the bottleneck buffer, the propagation delay, and the average packet size. PSQM, PAMS, and PESQ results are highly correlated with MOS [79]. Rix gave a mapping from PESQ score to MOS [80]. Galiotos et al. [81] used the E-model for VoIP QoS control. Ding and Goubran [82] studied the effect of delay and jitter on the E-model. Sun proposed a way to combine the PESQ test with the E-model [83]. A good review paper on perceptual QoS tests for VoIP is available in the literature [84].

Time synchronization is important for VoIP applications. Melvin and Murphy [85] suggested using the network time protocol (NTP [86]) and the global positioning system (GPS) to synchronize time for the VoIP QoS. Johannessen [87] gave a nice survey regarding the importance of time synchronization, the problem with current time clocks, the available standards, and the available time synchronization methods.

One of the major problems in time synchronization is clock skew. Paxson [88] studied this problem and proposed a way to determine its presence and remove it. Moon et al. [89] studied different estimation methods and proposed the use of linear programming (LP) to remove clock skew. Zhang et al. [90] proposed the convex lower bound method to replace the LP method. Bletsas suggested using the Kalman filtering method for the time synchronization [91].

The voice quality of VoIP applications and their sensitivity to delay, jitter and loss partially depend on the codec being used. There are many types of codecs, including Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM), Sub-Band Coding (SBC), Multi-Pulse Excited (MPE), Regular-Pulse Exited (RPE), Code-Excited Linear Predictive (CELP) etc. [92]. The three commonly used codecs, ITU G.711, G.729 and G.723, are PCM codecs. ITU G.721, G.726, G.727 are Adaptive Differential Pulse Code Modulation (ADPCM) codecs. ITU G.728 is a CELP codec. Speex, a publicly available tool, implements another CELP codec designed specifically for VoIP applications [93]. Global System for Mobile communications (GSM) has been used in cell phones.

Packet size also impacts VoIP. Oouch et al. [94] reported that for the no congestion case, large packets are more efficient. If congestion is present, then short packets are better. Scheets et al. [95] studied the relation between the targeted end-to-end delay and the packet size.

All of the QoS techniques discussed in the previous sections (Sec. B&C) can be implemented to support VoIP. Some good surveys on the QoS of VoIP are available [3, 42, 96]. Bilhaj and Mase [97] used admission control based on delay and loss rate for the QoS control of VoIP networks. Bolot and Vega-Garcia [98] used a jitter control and a combined error and rate control method to improve the VoIP QoS. In an IETF review of VoIP in public networks, Floyd and Kempf [99] recommended an

adaptive variable-bit-rate codec that is able to vary its bit rate according to estimates of congestion. Homayounfar [100] reviewed the Adaptive Multirate Codec (AMR) for mobile networks. Qiao et al. [101] combined adaptive rate control with priority marking and used the E-model for voice quality prediction. Tao et al. [102] improved VoIP quality through path switching. Besides source rate, playout rate can also impact VoIP voice quality. Sun and Ifeachor [103] used voice quality prediction from the E-model to adjust VoIP playout rate. Ranganathan and Kilmanrtin [104] used neural networks and fuzzy logic control to adapt the playout delay in VoIP networks.

Rate adaption needs network bandwidth estimation. Prasad [105] gave a survey on bandwidth estimation. Shriram et al. [106] present a comparison of the different bandwidth estimation methods.

E. Internet Models

The Internet is a large and heterogeneous network of networks. Its traffic is self-similar, statistically heavy-tailed distributed. It is growing and changing in drastic ways over time. Thus, modeling the Internet is a difficult task [107]. The models will be different for different research interests [108].

Test-beds such as the PlanetLab [109] are available for gathering "real-world" data. PlanetLab is an open, globally distributed platform for developing, deploying and accessing planetary scale network services. Network services deployed on PlanetLab experience most of the behavior of the real Internet. Spring et al. [110] discussed the current situations regarding PlanetLab.

Emulation tools such as the NIST Net [111] enable a local area network to emulate the behavior of a wide area network for research purpose. NIST Net enables a Linux PC to be set up as a router, then delay and drop packets according to a given

probability distribution.

Simulations are useful for understanding the dynamics, illustrating a point, or exploring unexpected behavior [107]. One of the most commonly used simulation tools is Network Simulator 2 (NS-2) [112].

Although the Internet is a packet-switched network, simulating its performance at the packet level is becoming increasingly difficult with its growing scale. Liu et al. [113] proposed using fluid model simulations instead. There are some fluid models available for AQM control [114]. Parlos and Ye [115] abstracted the Internet as a flow transport media characterized by time-varying time-delays and proposed a model, which expresses the flow arrival rate as a function of the flow departure rate and the time-varying transport time delay. Konstantinou [116] used fluid model in the study of source buffering control.

Modeling the networks using a white box model is very difficult. Ohsaki et al. [117] used black box approaches for modeling the dynamics of networks. Both linear and nonlinear methods can be used for modeling networks. Bremler-Barr et al. [118] used several kinds of linear predictors for predicting round trip time (RTT). Tao [119] used Markov model for path performance estimation. Hasegawa et al. [120] showed that it is worth trying nonlinear models for the Internet traffic prediction. Boné [121] gave a review on application of neural networks for multi-step-ahead time series prediction. Wang et al. [122] used radial basis function (RBF) neural networks for RTT prediction in QoS control. Kommaraju [123] reported that RBF predictors outperformed others in multi-step-ahead network accumulation prediction. Shah et al. [124] tried the same with both linear AutoRegressive (AR) and non-linear AutoRegressive (NAR) models. They reported that "the dynamic predictors fail to perform significantly better than the simple predictors over higher frequencies".

Good estimation and prediction of delay, loss, accumulation etc. of networks can

help to approach congestion control and QoS control in an end-to-end way, in a manner that is scalable. Various techniques have been used, including linear prediction, neuro-prediction, genetic algorithm etc. for delay prediction [125]. Doddi [126] used black box approached to model the delay of networks. Jiang and Schulzrinne [127] modeled the packet loss with a Gilbert model. Mehrvar and Soleymani [128] predicted the average loss rate with some traffic indicator. Roychoudhuri and Al-Shaer [129] predicted the possibility of packet loss for a period of time using the delay signal. Khariwal [130] studied prediction and control of packet accumulation within a network.

F. Control

Control theory has found applications in congestion control, AQM, and adaptive rate control as seen in previous sections. In AQM schemes, the system can be viewed as controlling an integrator via time-variant links. Bauer et al. [131] studied the stability conditions for such a system. To control the QoS over best-effort networks is to control a system with time-varying time-delay. The time-varying time-delay might be either modeled as a state delay or an input/output delay. There are many studies in the literature on the controllability and stability of such systems. The methods being used include LMI [132], Lyapunov functions [133], and stochastic processes [134], etc.. In most studies the time-varying part of the delay is modeled as an uncertainty in the delay to be overcome.

Predictive controls compensate for the delays instead of trying to tolerate them. Model-predictive control (MPC) [135, 136] is widely used in process industries, where almost constant but unknown time delay is a common phenomenon. An MPC controller uses a mathematical model of the process to predict the future effects of current

control actions. It is a receding horizon controller, as it normally applies the first of a series of calculated control actions and measures their results [135]. The reasons for widespread use of MPC are: explicit incorporation of process model, explicit handling of delays, future control horizon, and direct handling of constraints. A survey of the MPC controllers used in industry is given by Qin et al. [137].

CHAPTER III

SPEECH QUALITY EVALUATION

In this chapter, first the available speech quality evaluation methods are reviewed. Both subjective methods and objective methods are discussed. The formulae for mapping the objective scores to the subjective scores are given. Then the method for estimating E-model parameters for the Speex codec is discussed. The resulting parameters are given in this chapter.

A. Review of Available Speech Quality Evaluation Methods

The fundamental testing criterion for speech quality is the subjective quality test. The most widely used one is the mean opinion score (MOS). But the MOS test is expensive and time consuming, so objective tests have been developed to estimate subjective quality from the physical characteristics at the terminals. The objective tests can be classified into signal based methods and parameter based methods. The signal based methods need a reference signal, they are intrusive, but they are more accurate. The parameter based methods are non-intrusive, and are more appropriate for live monitoring and network planning [83, 84].

1. Subjective speech quality tests

The mean opinion score (MOS) is defined in ITU-T P.800 [74] as an absolute category rating (ACR) for the performance of the system under test. Listeners are asked to rate the quality of speech on a scale of 5 to 1, as shown in Table I. The arithmetic mean of all the opinion scores collected is the MOS [84].

The degradation mean opinion score (DMOS) is also defined by ITU-T P.800 [74]. It is a degradation category rating (DCR) for the performance of the system under

Table I. Opinion scale for MOS test [83].

Category	Speech Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table II. Opinion scale for DMOS test [83].

Score	Degradation level
5	Inaudible degradation
4	Audible degradation but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

test relative to the subjective quality of a reference system. Listeners are asked to compare the test speech to the original (reference) speech, and rate the degradation level on a scale of 5 to 1, as shown in Table II. The mean value of the results is called the DMOS [84].

ITU-T P.800 [74] has defined in detail the requirements for conducting the subjective tests. The MOS tests are normally required to be carried out under a controlled condition in a sound proof room. The subjective tests are expensive and time consuming [83].

2. Signal based objective speech quality tests

The signal based objective tests use two inputs signals, a reference (original) signal and the degraded (distorted) signal measured at the output of the system under test. They are intrusive, more accurate for measuring end-to-end speech quality, and not suitable for live monitoring.

The research of objective tests started with the use of signal-to-noise ratio (SNR) in the time domain, went into the spectral domain, and finally succeeded in the perceptual domain, which is based on the models of human auditory perception [84]. The Perceptual Speech Quality Measure (PSQM), which is based on Bark spectral distortion, was standardized as ITU-T P.861 [138] in 1998. The Perceptual Analysis Measurement System (PAMS) has used a different perceptual modeling than PSQM, and has also used a sophisticated time-alignment scheme. In order to come up with a test that is applicable to VoIP and mobile communications, the PAMS was compromised with PSQM+, which is an extension of PSQM, to become Perceptual Evaluation of Speech Quality (PESQ) [139], and this was standardized as ITU-T P.862 [140] in 2001.

PESQ is the ITU recommendation for objective speech quality assessment of 3.1 kHz (narrow-band) handset telephony and narrow-band speech codecs. It only measures the effects of one-way speech distortion and noise on speech quality. The effects of impairments related to two-way interaction are not reflected. PESQ compares an original signal with its degraded version, which is the result of passing through a communication system, and outputs a prediction of the MOS score of the degraded signal.

The first step of PESQ is to compute a series of delays between the original input and the degraded output for time alignment. The speech signal is divided into

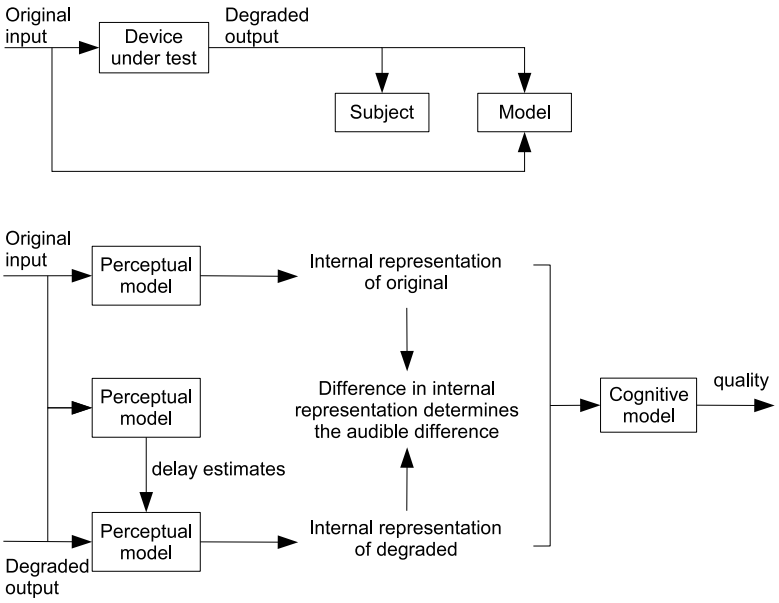

Fig. 1. Overview of basic philosophy used in PESQ [83].

time intervals each having a significant delay difference from the previous one. Then PESQ compares the original signal with the aligned, degraded output using a perceptual model as shown in Fig. 1. Both the original and degraded signals are transformed to an internal representation analogous to the psychophysical representation of audio signals in the human auditory system, which takes into account of perceptual frequency (Bark) and loudness (Sone). This is done by time alignment, level alignment to a calibrated listening level, time-frequency mapping, frequency warping, and compressive loudness scaling. The internal representation is processed to take care of effects that may have little perceptual significance. More severe effects are only partially compensated. Two error parameters are computed in the cognitive model and are combined to give an objective listening quality MOS. The details for this approach can be found in the C source code provided by ITU-T P.862 [140].

The PESQ score is between -0.5 and 4.5, while a absolute category rating (ACR)

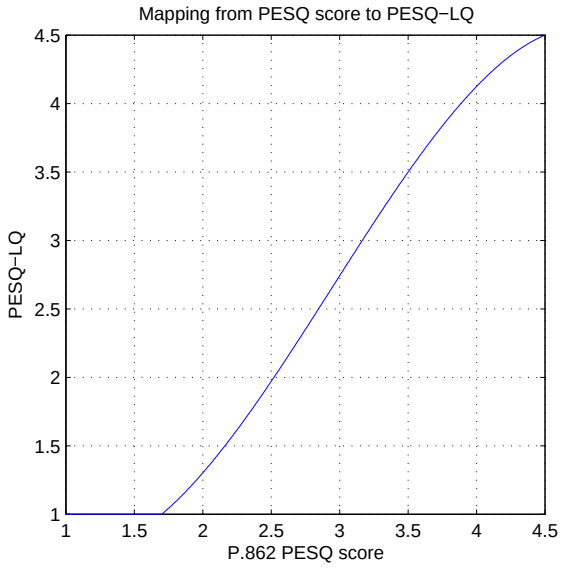

Fig. 2. Mapping from PESQ score to PESQ-LQ [83].

listening quality (LQ) mean opinion score (MOS) is on a $1 \sim 5$ scale. PESQ-LQ was proposed to map the P.862 PESQ score to an average P.800 ACR LQ MOS scale, in the range of 1 to 4.5. PESQ-LQ is defined as follows, where x is the P.862 PESQ score and y is the corresponding PESQ-LQ [83]:

$$y = \begin{cases} 1.0, & x \leq 1.7 \\ -0.157268x^3 + 1.386609x^2 - 2.504699x + 2.023345, & x > 1.7 \end{cases} . \tag{3.1}$$

The functional form of PESQ-LQ is shown in Fig. 2.

3. Parameter based objective speech quality tests

The parameter based methods do not need a reference signal and can be used for live monitoring and network planning. The E-model, standardized by ITU-T as Recommendation G.107 [75], is a parameter based method. It is based on the assumption

that the impairment factors are additive. It has 20 input parameters that represent the terminal, network, and environmental quality factor. Its output is called the R-value, which is between 0 and 100.

The R-value is calculated by:

$$R = R_0 - I_s - I_d - I_e + A, \tag{3.2}$$

where, R_0 represents the signal-to-nose ratio at the 0 dB point; I_s accounts for those impairments that occur simultaneously with speech; I_d represents the impairments caused by delay; I_e represents the impairments caused by losses, codecs, and packet loss concealment (PLC) etc.; and A represents advantage factors the users may accept to trade-off for bad voice quality [141].

A MOS score can be derived from the R-value by [142]:

$$MOS = \begin{cases} 1 & R \leq 0 \\ 1 + 0.035R + 7 \times 10^{-6} & \\ \quad \times R(R-60)(100-R) & 0 < R < 100 \\ 4.5 & R \geq 100 \end{cases}. \tag{3.3}$$

This MOS is called E-model MOS in this research. The mapping from R-value to MOS is shown in Fig. 3.

ITU G.107 has provided a set of default values for all the parameters in E-model for network planning. Using default values for all other factors except I_d and I_e, reduces the model to [102]

$$R = 94.2 - I_e - I_d. \tag{3.4}$$

The relation between I_e and the overall packet loss rate e (between 0 and 1) is given by [102]

$$I_e = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e), \tag{3.5}$$

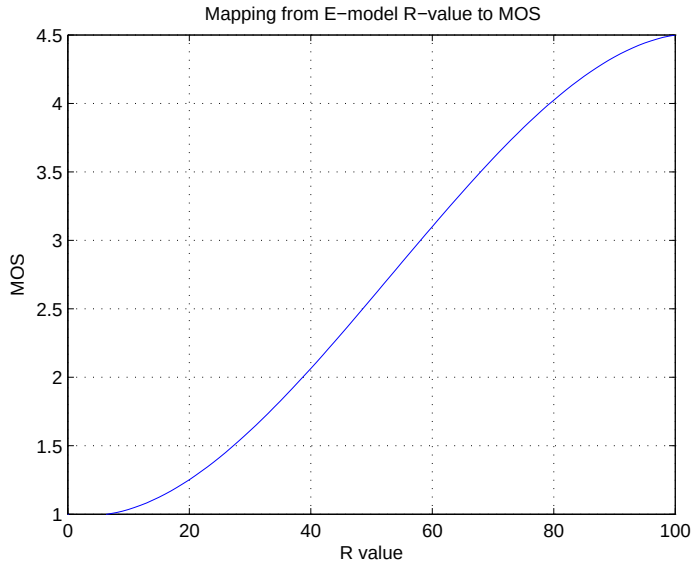

Fig. 3. Mapping from E-model R-value to MOS.

where γ_1, γ_2, and γ_3 are determined by the codec being used. The parameters for several codecs taken from [102] are repeated here in Table III.

The relation between I_d and the end-to-end delay d (in milliseconds) can be estimated by [102]

$$I_d = 0.024d + 0.11(d - 177.3)I(d - 177.3), \tag{3.6}$$

where,

$$I(x) = \begin{cases} 0, & x < 0 \\ 1, & \text{otherwise} \end{cases}.$$

B. Parameter Estimation of E-model for the Speex Codec

The E-model is a non-intrusive way of objectively estimating the speech quality of VoIP applications running on networks. The R-values of the E-model can be mapped

Table III. The values of γ_1, γ_2 and γ_3 for several codecs [102].

Codec	frames/packet	PLC	γ_1	γ_2	γ_3
G.723.1.B-5.3	1	silence	19	71.38	6
G.723.1.B-6.3	1	silence	15	90.00	5
G.729	1	silence	10	47.82	18
G.723.1.A+VAD-6.3	1	none	15	30.50	17
G.729A+VAD	2	none	11	30.00	16

to MOS using equation (3.3). The impairment factors are assumed to be additive in this model. In the simplified model, equation (3.4), the I_e term, which is related to the loss rate in the network, has a significant impact on the resulting R-value of the E-model. The relation between the loss rate and the I_e term is given by equation (3.5). But there is currently no ready made coefficients for this model for the Speex codec used in this research [93]. In this section the γ_1, γ_2, and γ_3 of equation (3.5) for the Speex codec are to be determined. As both the number of frames per packet and error concealment method can impact speech quality [143], in this research only the coefficients for two distinct cases are investigated. In the two cases under consideration, the speech is to be encoded by the Speex codec, and to be send at 100 ms interval and 20 ms interval, which correspond to 5 frames per packet and 1 frame per packet, respectively. The repeating-last-received-packet method is used for error concealment.

1. Method for estimating γ_1, γ_2, and γ_3

First, trace-files with random losses are generated. The loss rates are in the range from 0% to 20%. Then the original speech file is encoded by the Speex codec. For the 100 ms packet interval case, five frames are packetized into one packet. For the 20 ms

packet interval case, one frame is packetized as one packet. Then the packets are sent through an emulator which drops the packets according to the trace files. Thus the VoIP stream has experienced different loss rates. The received packets are then decoded to the degraded speech files using the repeating-last-received-packet error concealment method. The degraded speech files are compared to the original speech file using the PESQ test. The resulting PESQ scores are mapped to MOS using the PESQ-LQ method.

Once the MOSs are obtained, they can be mapped to the E-model R-values by inverting equation (3.3). The approximate inverse formula is [103]

$$R = 3.026x^3 - 25.314x^2 + 87.060x - 57.336, \tag{3.7}$$

where x is the MOS. As the PESQ does not reflect the effect of delays in the networks, the I_d value in equation (3.4) can be taken as 0. So the I_e value can be obtained as

$$I_e = 94.2 - R. \tag{3.8}$$

With sufficient pairs of loss rate e and impairment term I_e, the coefficients γ_1, γ_2 and γ_3 can be obtained by fitting a curve through the points formed by the (e, I_e) pairs on the I_e vs e plot. This can be done using the Gauss-Newton iteration method for non-linear parameter estimation [144].

The model of I_e can be given as

$$I_e[n] \;=\; s[n] + w[n], \tag{3.9}$$

where,

$$s[n] \;=\; \gamma_1 + \gamma_2 \ln\left(1 + \gamma_3 e[n]\right), \quad n = 0, 1, \ldots, N-1, \tag{3.10}$$

N is the total number of (e, I_e) pairs, and $w[n]$ is white Gaussian noise. To find γ_1, γ_2

and γ_3 using the least square estimation method one must minimize the cost function

$$J = \sum_{n=0}^{N-1} \left(I_e[n] - s[n]\right)^2. \tag{3.11}$$

When the loss rate e is given in the range of $0 \sim 1$, the initial value of γ_3 can be taken as $\gamma_{3,0} = 100$. Then the model of I_e can be given as

$$\begin{aligned} I_e[n] &= s[n] + w[n], \\ s[n] &= \begin{bmatrix} 1 & \ln\left(1 + 100e[n]\right) \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}, \\ & n = 0, 1, \ldots, N-1. \end{aligned} \tag{3.12}$$

In vector form

$$X = S + W, \tag{3.13}$$

where,

$$X = \begin{bmatrix} I_e[0] \\ \vdots \\ I_e[N-1] \end{bmatrix}, \tag{3.14}$$

$$W = \begin{bmatrix} w[0] \\ \vdots \\ w[N-1] \end{bmatrix}, \tag{3.15}$$

$$S = H\Gamma, \tag{3.16}$$

with

$$H = \begin{bmatrix} 1 & \ln(1+100e[0]) \\ \vdots & \vdots \\ 1 & \ln(1+100e[N-1]) \end{bmatrix}, \tag{3.17}$$

$$\Gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}. \tag{3.18}$$

The least square solution, which minimizes equation (3.11) when $\gamma_{3,0} = 100$, can be obtained by

$$\Gamma_0 = \begin{bmatrix} \gamma_{1,0} \\ \gamma_{2,0} \end{bmatrix} = \left(H^T H\right)^{-1} H^T X. \tag{3.19}$$

Thus the initial coefficients

$$\theta_0 = \begin{bmatrix} \gamma_{1,0} \\ \gamma_{2,0} \\ \gamma_{3,0} \end{bmatrix} \tag{3.20}$$

for the Gauss-Newton iteration method is obtained.

Given θ_0, the linearized model around that point is given by

$$\begin{aligned} s[n;\theta] &= s[n;\theta_0] + (\gamma_1 - \gamma_{1,0}) + \ln(1+\gamma_{3,0})(\gamma_2 - \gamma_{2,0}) \\ &\quad + \frac{\gamma_{2,0} e[n]}{1+\gamma_{3,0} e[n]}(\gamma_3 - \gamma_{3,0}) \\ &= s[n;\theta_0] - \gamma_{1,0} - \ln(1+\gamma_{3,0})\gamma_{2,0} - \frac{\gamma_{2,0} e[n]}{1+\gamma_{3,0} e[n]}\gamma_{3,0} \\ &\quad + \gamma_1 + \ln(1+\gamma_{3,0})\gamma_2 + \frac{\gamma_{2,0} e[n]}{1+\gamma_{3,0} e[n]}\gamma_3 \\ &= s[n;\theta_0] - h[n;\theta_0]\theta_0 + h[n;\theta_0]\theta, \end{aligned} \tag{3.21}$$

where,

$$h[n;\theta_0] = \left[\begin{array}{ccc} 1 & \ln(1+\gamma_{3,0}e[n]) & \dfrac{\gamma_{2,0}e[n]}{1+\gamma_{3,0}e[n]} \end{array}\right]. \tag{3.22}$$

In matrix form this can be written as

$$S[\theta] = S[\theta_0] - H[\theta_0]\theta_0 + H[\theta_0]\theta, \tag{3.23}$$

where,

$$H[\theta_0] = \left[\begin{array}{ccc} 1 & \ln(1+\gamma_{3,0}e[0]) & \dfrac{\gamma_{2,0}e[0]}{1+\gamma_{3,0}e[0]} \\ \vdots & \vdots & \vdots \\ 1 & \ln(1+\gamma_{3,0}e[N-1]) & \dfrac{\gamma_{2,0}e[N-1]}{1+\gamma_{3,0}e[N-1]} \end{array}\right], \tag{3.24}$$

and

$$J = (X - S[\theta_0] + H[\theta_0]\theta_0 - H[\theta_0]\theta)^T(X - S[\theta_0] + H[\theta_0]\theta_0 - H[\theta_0]\theta), \tag{3.25}$$

and

$$\hat{\theta} = \theta_0 + (H^T[\theta_0]H[\theta_0])^{-1}H^T[\theta_0](X - S[\theta_0]). \tag{3.26}$$

The iteration algorithm is

$$\theta_{k+1} = \theta_k + (H^T[\theta_k]H[\theta_k])^{-1}H^T[\theta_k](X - S[\theta_k]), \tag{3.27}$$

$$S[\theta_k] = \left[\begin{array}{c} \gamma_{1,k} + \gamma_{2,k}\ln(1+\gamma_{3,k}e[0]) \\ \vdots \\ \gamma_{1,k} + \gamma_{2,k}\ln(1+\gamma_{3,k}e[N-1]) \end{array}\right], \tag{3.28}$$

$$H[\theta_k] = \left[\begin{array}{ccc} 1 & \ln(1+\gamma_{3,k}e[0]) & \dfrac{\gamma_{2,k}e[0]}{1+\gamma_{3,k}e[0]} \\ \vdots & \vdots & \vdots \\ 1 & \ln(1+\gamma_{3,k}e[N-1]) & \dfrac{\gamma_{2,k}e[N-1]}{1+\gamma_{3,k}e[N-1]} \end{array}\right]. \tag{3.29}$$

Convergence is obtained when

$$\| \theta_{k+1} - \theta_k \|_2 < \epsilon, \tag{3.30}$$

where $0 < \epsilon \ll 1$ is a given positive small number. The final converged values are the estimates of the γ_1,γ_2, and γ_3 coefficients. Using these estimated coefficients $\hat{\gamma}_1$,$\hat{\gamma}_2$, and $\hat{\gamma}_3$, equation (3.5), equation (3.4) with $I_d = 0$, and equation (3.3), the estimated E-model MOS can be obtained.

Correlation coefficient ρ_{I_e} between the measured loss impairment factors I_e and the estimated loss impairment factors $\hat{I}_e$, root mean square error σ_{I_e} between the measured loss impairment factors I_e and the estimated loss impairment factors $\hat{I}_e$, and root mean square error σ_{MOS} between the measured MOS and the estimated MOS are used to evaluate the fitness of the estimates.

The correlation coefficient between the measured and estimated loss impairment factors is given by

$$\rho_{I_e} = \frac{\sum_{i=0}^{N-1} \left((I_{e,i} - \bar{I}_e)(\hat{I}_{e,i} - \bar{\hat{I}}_e) \right)}{(N-1)s(I_e)s(\hat{I}_e)}, \tag{3.31}$$

where, $\{I_{e,i} | i = 0, \ldots, N-1\}$ are the measured loss impairment factors from PESQ-LQ test, $\{\hat{I}_{e,i} | i = 0, \ldots, N-1\}$ are the estimated loss impairment factors using equation (3.5) and the estimated $\hat{\gamma}_1$, $\hat{\gamma}_2$, and $\hat{\gamma}_3$; $\bar{I}_e$ and $\bar{\hat{I}}_e$ are the means of $\{I_{e,i} | i = 0, \ldots, N-1\}$ and $\{\hat{I}_{e,i} | i = 0, \ldots, N-1\}$, respectively; $s(I_e)$ and $s(\hat{I}_e)$ are the standard deviation of $\{I_{e_i} | i = 0, \ldots, N-1\}$ and $\{\hat{I}_{e,i} | i = 0, \ldots, N-1\}$, respectively.

The root mean square error between the measured and estimated loss impairment factors is defined as

$$\sigma_{I_e} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \left(\hat{I}_{e,i} - I_{e,i} \right)^2}, \tag{3.32}$$

where,$\{I_{e,i} | i = 0, \ldots, N-1\}$ are the measured loss impairment factors from PESQ-LQ test, and $\{\hat{I}_{e,i} | i = 0, \ldots, N-1\}$ are the estimated loss impairment factors using

equation (3.5) and the estimated $\hat{\gamma}_1$, $\hat{\gamma}_2$, and $\hat{\gamma}_3$.

The root mean square error between the measured and estimated loss impairment factors is defined as

$$\sigma_{MOS} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(\hat{x}_i - x_i)^2}, \tag{3.33}$$

$\{x_i | i = 0, \ldots, N-1\}$ are the measured MOS from PESQ-LQ test, and $\{\hat{x}_i | i = 0, \ldots, N-1\}$ are the estimated E-model MOS using the estimated coefficients $\hat{\gamma}_1$, $\hat{\gamma}_2$, and $\hat{\gamma}_3$, equation (3.5), equation (3.4) with $I_d = 0$, and equation (3.3).

2. Parameter estimation results

a. 100 millisecond packet send interval

In this case, the original speech is encoded with the Speex codec into VoIP packets, which are sent every 100 ms . There are five frames per packet. The VoIP packets are sent through a network emulator, which gives random losses ranging from 0% to 20%. The received packets are then decoded to obtain the degraded speech using the repeating-last-received-packet error concealment method. The degraded speech is compared to the original speech for quality test using the PESQ method. The result is mapped to MOS using PESQ-LQ method. The MOSs of the speech vs the loss rates are plotted in Fig. 4. The corresponding R-values of the speech vs the loss rates are plotted in Fig. 5. The corresponding I_e values vs the loss rates are plotted in Fig. 6. The initial estimates of the coefficients from equation (3.19) are $\gamma_{1,0} = 12.10$, $\gamma_{2,0} = 15.61$, $\gamma_{3,0} = 100$. The final estimates of the coefficients through the iteration algorithm using equation (3.27) through (3.29) are $\hat{\gamma}_1 = 17.24$, $\hat{\gamma}_2 = 40.14$, $\hat{\gamma}_3 = 12.02$. The estimated I_e values vs the loss rates using the initial and final estimation of the coefficients are plotted in Fig. 6. The corresponding estimated R-values vs the loss rates are plotted in Fig. 5. The corresponding estimated E-model MOSs vs the

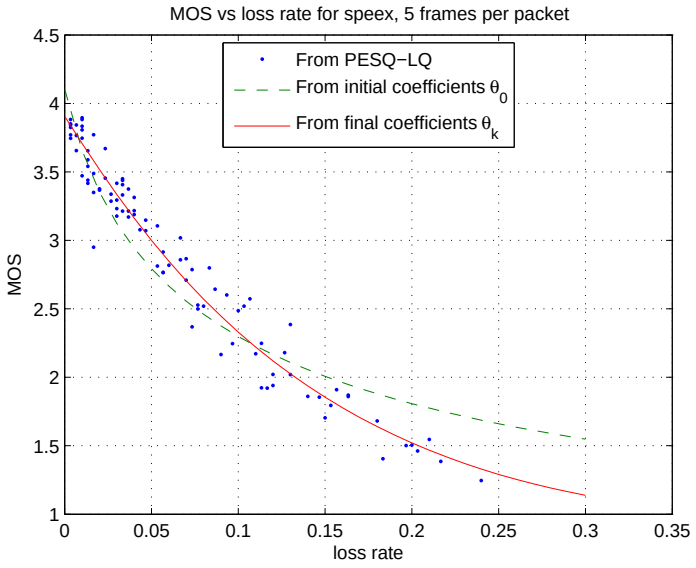

Fig. 4. MOS estimation of Speex with five frames per packet.

loss rates are plotted in Fig. 4. The correlation coefficient of the I_e estimation is $\rho_{I_e} = 0.9798$. The root mean square error of the I_e estimation is $\sigma_{I_e} = 3.1818$. The root mean square error of the MOS estimation is $\sigma_{MOS} = 0.15$. The results show that these estimates are good and have reasonable accuracy.

b. 20 millisecond packet send interval

In this case, the original speech is encoded with the Speex codec into VoIP packets, which are sent every 20 ms. There is one frame per packet. The other procedures are the same as in the previous case. The MOSs of the speech vs the loss rates are plotted in Fig. 7. The corresponding R-values of the speech vs the loss rates are plotted in Fig. 8. The corresponding I_e values vs the loss rates are plotted in Fig. 9. The initial estimates of the coefficients from equation (3.19) are $\gamma_{1,0} = 12.78$, $\gamma_{2,0} = 17.29$, $\gamma_{3,0} = 100$. The final estimates of the coefficients through the iteration algorithm using

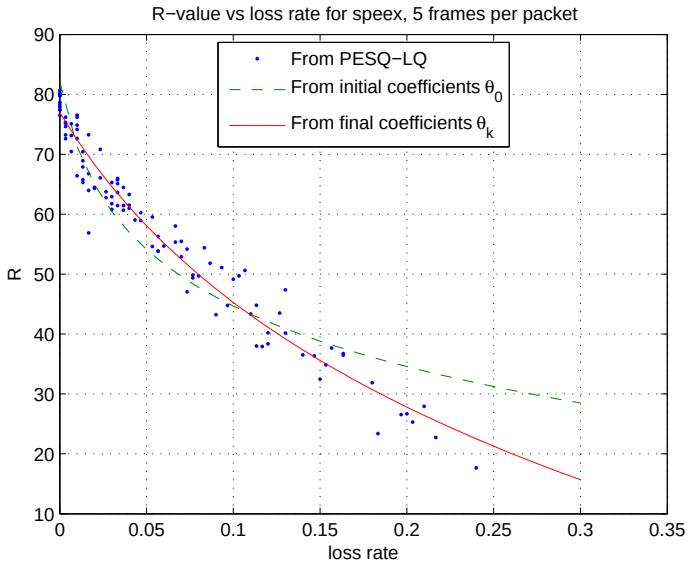

Fig. 5. R-value estimation of Speex with five frames per packet.

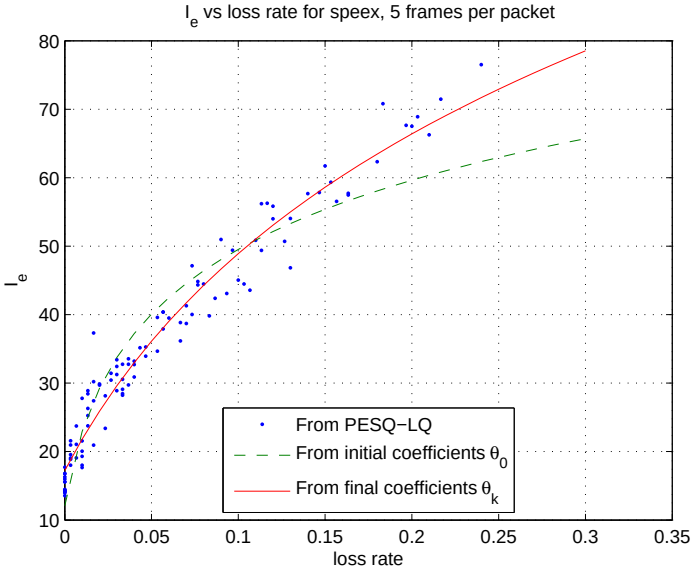

Fig. 6. I_e estimation of Speex with five frames per packet.

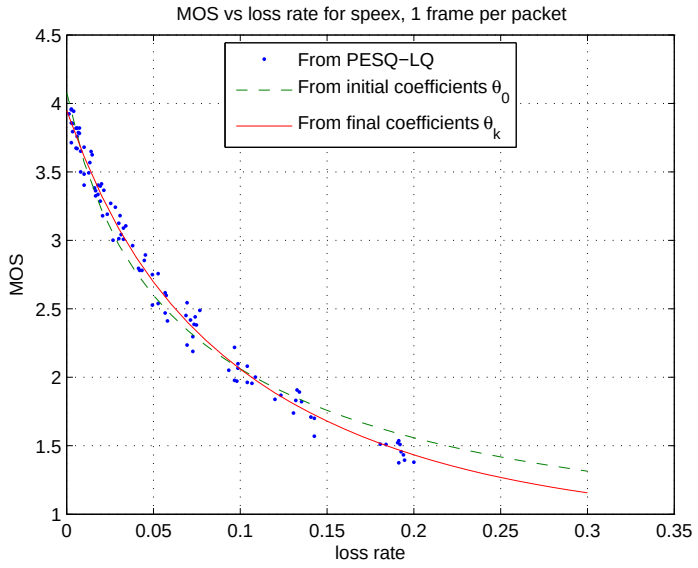

Fig. 7. MOS estimation of Speex with one frame per packet.

equation (3.27) through (3.29) are $\hat{\gamma}_1 = 16.19$, $\hat{\gamma}_2 = 24.91$, $\hat{\gamma}_3 = 36.17$. The estimated I_e values vs the loss rates using the initial and final estimation of the coefficients are plotted in Fig. 9. The corresponding estimated R-values vs the loss rates are plotted in Fig. 8. The corresponding estimated E-model MOSs vs the loss rates are plotted in Fig. 7. The correlation coefficient of the I_e estimation is $\rho_{I_e} = 0.9928$. The root mean square error of the I_e estimation is $\sigma_{I_e} = 1.9979$. The root mean square error of the MOS estimation is $\sigma_{MOS} = 0.09$. The results show that these estimates are good and have reasonable accuracy.

The final estimates of γ_1, γ_2 and γ_3 for the above two packet send intervals are presented in Table IV.

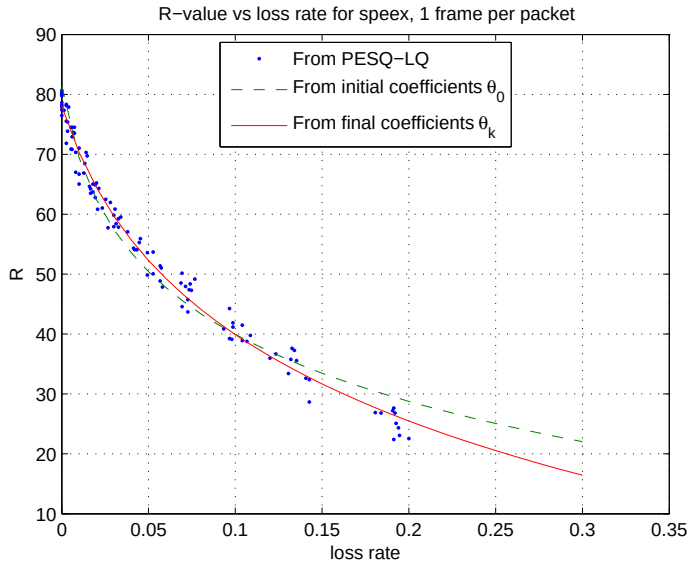

Fig. 8. R-value estimation of Speex with one frame per packet.

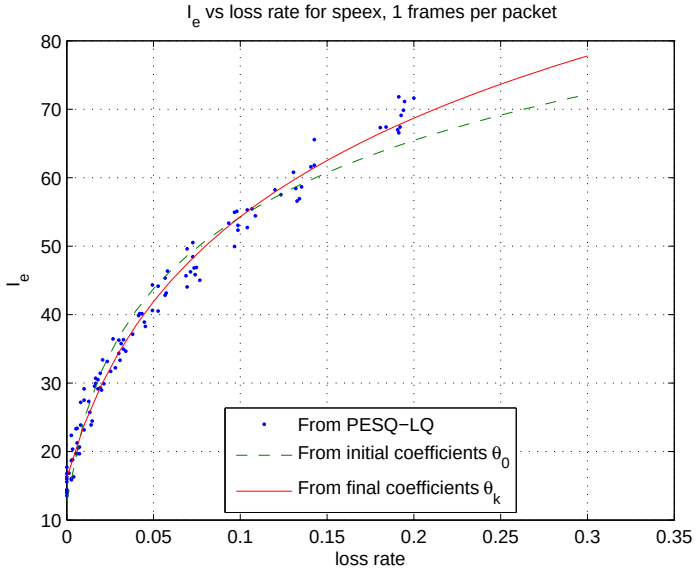

Fig. 9. I_e estimation of Speex with one frame per packet.

Table IV. The values of γ_1, γ_2 and γ_3 for Speex codec.

Codec	frames/packet	γ_1	γ_2	γ_3	ρ	σ_{I_e}	σ_{MOS}
Speex	5	17.24	40.13	12.02	0.9798	3.18	0.15
Speex	1	16.19	24.91	36.17	0.9928	2.00	0.09

C. Chapter Summary

In this chapter, both the available subjective speech quality evaluation methods and the available objective speech quality evaluation methods have been reviewed. The equations for mapping the objective scores to the subjective scores are also given. The Gauss-Newton iteration method has been used for estimating E-model parameters for the Speex codec used in this research. The resulting coefficients for calculating the I_e parameter in the E-model of Speex codec has been presented in Table IV.

CHAPTER IV

STUDY OF FLUID MODEL BASED SINGLE FLOW CONTROL

As stated in the problem definition, the target of this research is to find an end-to-end middleware based solution to improve the QoS for real-time multimedia applications in public best-effort networks. By doing so, this solution can be readily applied to any real-time application without accessing traffic information from the routers.

In a previous study from this group, Konstantinou [116] has looked into the possibility of using a fluid model for QoS control. Khariwal [130] tried to implement the model predictive control (MPC) method for QoS control using a packet based approach. In these studies, the playback buffer underrun was prevented by controlling the send rate, either reactively or predictively, through a send side buffer.

In this chapter, the method of using a fluid model and a controlled send side buffer to improve the QoS for real-time multimedia applications is investigated.

A. Assumptions and Concepts in Fluid Model Based Single Flow Control

Best-effort networks are packet switching networks. It is natural to build packet-level models for them. But with increasing size and complexity it becomes more efficient to model the network as a fluid pipe system [145] and model it with a fluid model [113].

Assume that the flow transported on a network has experienced no losses. Then the flow will be experiencing a continuous time-varying time-delay in the network. The time-varying time-delay has a positive lower bound and a finite upper-bound. Under the no losses assumption, it is a conservative system, where every bit of the flow put into the system will eventually come out at the other end. If this assumption is violated, the lost packets will be associated with infinite delays, and the following fluid models cannot be obtained.

It is also assumed that the flow of interest is a relatively small amount of the flow compared to all the other flows in the network. Thus the network condition is dominated by the cross flow. Under this assumption, for a given cross flow condition, the time-varying time-delay in the network experienced by that single flow is a function of time, it is independent of the flow rate of the single flow of interest. Thus all flow leaving the source at the same time will experience the same delay. If this assumptions is violated, the flow rate of the single flow of interest will change the delay it will experience. In that case, the following fluid models will not reflect all the dynamics of the system. In this case, whether a controlled send side buffer can improve the QoS for real-time multimedia applications is still an open question.

If packet A enters the network before packet B, but packet B leaves the network before packet A, then there is a packet reordering in the network. Similarly, in a fluid system if at the entering point a piece of flow tinted with a drop of red ink enters the system before another piece of flow tinted with a drop of blue ink, but at the exit the piece of flow tinted with the blue ink leaves before the piece of flow tinted with red ink, then there is flow reversal in the fluid system. Flow reversal can happen when there are more than one path in the fluid system from the source to the destination. It is the same for networks system that flow reversal can happen when there are multiple network paths from the source to the destination inside the networks.

B. A Fluid Model without Flow Reversal

If a piece of the flow enters the system at the departure time t_d, and leaves the system at the arrival time t_a, then the delay measurement can either be associated with the departure as $\tau_d(t_d)$ or associated with the arrival time as $\tau_a(t_a)$. The relation between

the departure time and the arrival time can be expressed as:

$$t_a = t_d + \tau_d(t_d), \tag{4.1}$$

or,

$$t_a - \tau_a(t_a) = t_d. \tag{4.2}$$

Lemma 1 *If there is no flow reversal, then*

$$\dot{\tau}_d(t_d) > -1, \qquad or, \tag{4.3}$$

$$\dot{\tau}_a(t_a) < 1. \tag{4.4}$$

Proof: Consider two pieces of flow, which enter the system at departure time t_{d1} and t_{d2}, where $t_{d1} < t_{d2}$, and leave the system at arrival time t_{a1} and t_{a2}, respectively. When there is no flow reversal, the flow entering the system must keep its order when leaving the system. So $t_{a1} < t_{a2}$ must be true. For the delay expressed in terms of departure time, it can be written as:

$$\begin{aligned}
t_{a1} < t_{a2} &\Rightarrow t_{d1} + \tau_d(t_{d1}) < t_{d2} + \tau_d(t_{d2}) \\
&\Rightarrow \tau_d(t_{d1}) - \tau_d(t_{d2}) < t_{d2} - t_{d1} \\
&\Rightarrow \frac{\tau_d(t_{d1}) - \tau_d(t_{d2})}{t_{d2} - t_{d1}} < 1 \\
&\Rightarrow -\lim_{t_{d2} \to t_{d1}} \frac{\tau_d(t_{d2}) - \tau_d(t_{d1})}{t_{d2} - t_{d1}} < 1 \\
&\Rightarrow \dot{\tau}_d(t_{d1}) > -1.
\end{aligned} \tag{4.5}$$

For the delay expressed in terms of arrival time, it can be written as:

$$\begin{aligned}
t_{d1} < t_{d2} &\Rightarrow t_{a1} - \tau_a(t_{a1}) < t_{a2} - \tau_a(t_{a2}) \\
&\Rightarrow \tau_a(t_{a2}) - \tau_a(t_{a1}) < t_{a2} - t_{a1} \\
&\Rightarrow \frac{\tau_a(t_{a2}) - \tau_a(t_{a1})}{t_{a2} - t_{a1}} < 1
\end{aligned}$$

$$\Rightarrow \lim_{t_{a2}\to t_{a1}} \frac{\tau_a(t_{a2}) - \tau_a(t_{a1})}{t_{a2} - t_{a1}} < 1$$
$$\Rightarrow \dot{\tau}_a(t_{a1}) < 1. \tag{4.6}$$

■

In many publications, by default the delay is given in terms of arrival time, e.g. see [114, 146]. As a result the derivative of the delay is required to be less than one, e.g. see [36, 147].

Assume that the send rate at the source is $u(t_d)$ and it is continuous, then the cumulative sending flow $U(t_d)$ can be defined as

$$U(t_d) \triangleq \int_0^{t_d} u(t)\mathrm{d}t. \tag{4.7}$$

Assume that the arrival rate at the destination is $z(t_a)$ and it is also continuous, then the cumulative arriving flow $Z(t_a)$ can be defined as

$$Z(t_a) \triangleq \int_0^{t_a} z(t)\mathrm{d}t. \tag{4.8}$$

If there is no flow reversal, i.e. the order of the flow is preserved, then the relation between the cumulative flows can be given in terms of the departure time as

$$Z[t_d + \tau_d(t_d)] = U(t_d), \tag{4.9}$$

or, in terms of the arrival time as

$$Z(t_a) = U[t_a - \tau_a(t_a)]. \tag{4.10}$$

Theorem 1 *Given the delay in terms of the departure time the relation between the send rate and the arrival rate is given by*

$$z(t_a) = \frac{u(t_d)}{1 + \dot{\tau}_d(t_d)}. \tag{4.11}$$

Given the delay in terms of the arrival time the relation between the send rate and the arrival rate is given by

$$z(t_a) = u(t_d)[1 - \dot{\tau}_a(t_a)]. \tag{4.12}$$

Proof: Taking derivatives on both sides of equation (4.9) and using the chain rule results in

$$\begin{aligned} \frac{\mathrm{d}Z[t_d + \tau_d(t_d)]}{\mathrm{d}t_d} = \frac{\mathrm{d}U(t_d)}{\mathrm{d}t_d} \quad &\Rightarrow \quad \frac{\mathrm{d}Z(t_a)}{\mathrm{d}t_a}\frac{\mathrm{d}[t_d + \tau_d(t_d)]}{\mathrm{d}t_d} = \dot{U}(t_d) \\ &\Rightarrow \quad \dot{Z}(t_a)[1 + \dot{\tau}_d(t_d)] = \dot{U}(t_d) \\ &\Rightarrow \quad \dot{Z}(t_a) = \frac{\dot{U}(t_d)}{1 + \dot{\tau}_d(t_d)} \\ &\Rightarrow \quad z(t_a) = \frac{u(t_d)}{1 + \dot{\tau}_d(t_d)} \end{aligned} \tag{4.13}$$

If the delay is expressed in terms of the arrival time, taking derivatives on both sides of equation (4.10) and using the chain rule results in

$$\begin{aligned} z(t_a) = \dot{Z}(t_a) &= \frac{\mathrm{d}U[t_a - \tau_a(t_a)]}{\mathrm{d}t_a} \\ &= \dot{U}(t_d)\frac{\mathrm{d}[t_a - \tau_a(t_a)]}{\mathrm{d}t_a} \\ &= \dot{U}(t_d)[1 - \dot{\tau}_a(t_a)] \\ &= u(t_d)[1 - \dot{\tau}_a(t_a)]. \end{aligned} \tag{4.14}$$

■

Thus if there is no flow reversal, the relation between the send rate and the arrival rate can be expressed with either the delay given in terms of the departure time or the delay given in terms of the arrival time. The expression with the delay given in terms of the arrival time is the more commonly used one in the literature.

C. A Fluid Model with Flow Reversal

If there is flow reversal in the system, different parts of the flow which enter the system at different times at the source might arrive at the destination at the same time. In that case it is not possible to associate a single delay value to that arrival time. The delay given in terms of the arrival time becomes a multivalued function. It would be more convenient then to express the delay in terms of the departure time.

Assume that t_0 is the starting time, such that

$$u(t_d) = 0, \qquad U(t_d) = 0, \qquad \text{for } t_d < t_0. \tag{4.15}$$

Define an auxiliary function

$$f_{t_a}(t_d) \triangleq t_d + \tau_d(t_d) - t_a. \tag{4.16}$$

The solutions to the inequality

$$f_{t_a}(t_d) \leq 0, \qquad \text{where } t_d \geq t_0, \tag{4.17}$$

are the departure time intervals, during which the flow entering the system will arrive at the destination by time t_a. Under the assumption of a conservative system, the relation between the cumulative sending flow and the cumulative arriving flow is [115]:

$$Z(t_a) = \int_{t_0}^{t_a} z(t)\mathrm{d}(t) = \int_{t_0}^{t_a} \omega(\phi, t_a)u(\phi)\mathrm{d}\phi, \tag{4.18}$$

where the weight function $\omega(\phi, t_a)$ is defined as

$$\omega(\phi, t_a) = \begin{cases} 1, & f_{t_a}(\phi) \leq 0 \\ 0, & f_{t_a}(\phi) > 0 \end{cases} \tag{4.19}$$

Lemma 2 *The following expression is true:*

$$\frac{\mathrm{d}t_a}{\mathrm{d}t_d} = 1 + \dot{\tau}_d(t_d), \tag{4.20}$$

and if $\tau_d(t_d) \neq -1$

$$\frac{\mathrm{d}t_d}{\mathrm{d}t_a} = \frac{1}{1 + \dot{\tau}_d(t_d)}. \tag{4.21}$$

Proof: Taking the derivatives on both sides of equation (4.1) in respect to t_d, equation (4.20) can be obtained. Under the condition $\tau_d(t_d) \neq -1$, taking the reciprocal on both sides of equation (4.20), equation (4.21) is obtained. ■

If the solution to equation (4.17) is not the null set, then assume it has the following form:

$$t_d \in [t_{d_0}, t_{d_1}] \cup [t_{d_2}, t_{d_3}] \cup \ldots \cup [t_{d_{2M}}, t_{d_{2M+1}}] \cup \{\tilde{t}_{d_1}\} \cup \ldots \cup \{\tilde{t}_{d_N}\} \tag{4.22}$$

where, $\{t_{d_i} | i = 1, \ldots, 2M+1\}$ and $\{\tilde{t}_{d_j} | j = 1, \ldots, N\}$ are solutions to

$$f_{t_a}(t_d) = 0, \tag{4.23}$$

where, t_{d_0} can be either a solution to equation (4.23) or equal to t_0, and all the subsets have no common element with each other.

Theorem 2 *The relation between the arrival rate and the departure rate for a single flow characterized by the delay function is given by*

$$z(t_a) = \begin{cases} 0, & \textit{if equation (4.23) has no root} \\ \sum_{i=0}^{2M+1} (-1)^{i+1} \frac{u(t_{d_i})}{1+\dot{\tau}_d(t_{d_i})}, & \dot{\tau}_d(t_{d_i}) \neq -1 \\ \infty, & \dot{\tau}_d(t_{d_i}) = -1 \end{cases}. \tag{4.24}$$

Proof: If equation (4.23) has no root, the solution to equation (4.17) is the null set, no flow arrives at the destination, and the arrival rate is $z(t_a) = 0$.

Otherwise, given the solutions (4.22), the flow relation becomes

$$\begin{aligned} Z(t_a) = \int_{t_0}^{t_a} z(t)\mathrm{d}(t) &= \sum_{i=0}^{M} \int_{t_{d_{2i}}}^{t_{d_{2i+1}}} u(\phi)\mathrm{d}\phi + \sum_{j=1}^{N} \int_{\tilde{t}_{d_j}}^{\tilde{t}_{d_j}} u(\varphi)\mathrm{d}\varphi \\ &= \sum_{i=0}^{M} \left[U(t_{d_{2i+1}}) - U(t_{d_{2i}})\right]. \end{aligned} \tag{4.25}$$

If for none of the solutions $\{t_{d_i}|i = 1, \ldots, 2M+1\}$ to equation (4.23), $\dot{\tau}_d(t_{d_j}) = -1$, then taking the derivatives on both sides of equation (4.25) and use equation (4.21) results in

$$\begin{aligned} z(t_a) = \frac{\mathrm{d}Z(t_a)}{\mathrm{d}t_a} &= \sum_{i=0}^{M} \left[u(t_{d_{2i+1}}) \cdot \frac{\mathrm{d}t_{d_{2i+1}}}{\mathrm{d}t_a} - u(t_{d_{2i}}) \cdot \frac{\mathrm{d}t_{d_{2i}}}{\mathrm{d}t_a}\right] \\ &= \sum_{i=0}^{M} \left[\frac{u(t_{d_{2i+1}})}{1 + \dot{\tau}_d(t_{d_{2i+1}})} - \frac{u(t_{d_{2i}})}{1 + \dot{\tau}_d(t_{d_{2i}})}\right] \\ &= \sum_{j=0}^{2M+1} (-1)^{j+1} \frac{u(t_{d_j})}{1 + \dot{\tau}_d(t_{d_j})}. \end{aligned} \tag{4.26}$$

If for some of the solutions t_{d_i} to equation (4.23), $\dot{\tau}_d(t_{d_j}) = -1$, then the arrival rate value $z(t_a)$ is infinite. ■

D. Discrete Time Fluid Model

To discretize the relation between the input flow and the output flow, the assumed form of the input flow entering the system is needed. Two cases are studied here. In the first case, the input flow is assumed to be the output of a controller which dictates the cumulative flow. In this case, the cumulative flow is the result of a discrete time signal passed through a zero-order-holder (ZOH). In the second case, the input flow is assumed to be the output of a controller which dictates the flow rate. In that case, the input flow rate is the result of a discrete time signal going through a ZOH.

1. Zero-order-hold on cumulative input flow

In this case the cumulative input flow is the result of a discrete time signal going through a ZOH. Assuming that the sampling period of the system is T, the discrete time version of the cumulative input flow is

$$U(k) = U(kT). \tag{4.27}$$

Assume that the delay of the network is bounded as follows

$$0 < T_{min} \leq \tau_d(kT) \leq T_{max} < \infty. \tag{4.28}$$

The discretized version of delay is

$$\tau_d(k) = \frac{\tau_d(kT)}{T}, \tag{4.29}$$

and is bounded as

$$l_{min} \leq \tau_d(k) \leq l_{max}, \qquad \text{where,} \quad l_{min} = \left\lfloor \frac{T_{min}}{T} \right\rfloor, l_{max} = \left\lceil \frac{T_{max}}{T} \right\rceil. \tag{4.30}$$

The auxiliary function equation (4.16) becomes

$$f_{kT}(lT + mT) = lT + mT + (1 - m)\tau_d(lT) + m\tau_d[(l + 1)T] - kT, \tag{4.31}$$

where, $l \in \mathbb{N}$, the set of natural numbers, $0 \leq m < 1$. If the delay is known analytically then the delay at time $lT + mT$ can be calculated directly, but because in reality the delay measurement are obtained at discrete intervals, so the delay at time $lT + mT$ is linearly interpolated from the measurements of time lT and $(l + 1)T$. The discrete time version of equation (4.31) is

$$f_k(l + m) = l + m + (1 - m)\tau_d(l) + m\tau_d(l + 1) - k. \tag{4.32}$$

Lemma 3 *All of the solutions of*

$$f_k(l+m) = 0 \tag{4.33}$$

are bounded by

$$k - l_{max} \leq l + m \leq k - l_{min}. \tag{4.34}$$

Proof: Using equation (4.30) and equation (4.32) in equation (4.33) results in

$$\begin{aligned} f_k(l+m) = 0 \quad &\Rightarrow \quad l + m + (1-m)\tau_d(l) + m\tau_d(l+1) - k = 0 \\ &\Rightarrow \quad l + m = k - (1-m)\tau_d(l) - m\tau_d(l+1) \\ &\Rightarrow \quad k - (1-m)l_{max} - ml_{max} \leq l + m \leq k - (1-m)l_{min} - ml_{min} \\ &\Rightarrow \quad k - l_{max} \leq l + m \leq k - l_{min}. \end{aligned} \tag{4.35}$$

■

Assume $t_0 = 0$, then equation (4.18) becomes

$$Z(k) = \int_0^{kT} z(t)\mathrm{d}(t) = \int_0^{kT} \omega(\phi, kT)u(\phi)\mathrm{d}\phi, \tag{4.36}$$

where

$$\omega(\phi, kT) = \begin{cases} 1, & f_k(\frac{\phi}{T}) \leq 0 \\ 0, & f_k(\frac{\phi}{T}) > 0 \end{cases}. \tag{4.37}$$

Theorem 3 *If for a single flow the cumulative input flow is the output of a ZOH, then the relation between the cumulative input flow and the cumulative output flow is given by*

$$Z(k) = \sum_{i=l_{min}}^{l_{max}+1} A_i(k)U(k-i), \tag{4.38}$$

where,

$$A_i(k) = F_k(k-i) - F_k(k-i+1), \tag{4.39}$$

with the auxiliary function defined as

$$F_k(l) = \begin{cases} 1, & f_k(l) \leq 0 \\ 0, & f_k(l) > 0 \end{cases} . \tag{4.40}$$

Proof: Solving the inequality

$$f_k(l+m) \leq 0 \tag{4.41}$$

gives a solution set in the form of

$$\begin{aligned} &[l_0+m_0, l_1+m_1] \cup [l_2+m_2, l_3+m_3] \cup \ldots \\ &\cup [l_{2M}+m_{2M}, l_{2M+1}+m_{2M+1}] \cup \{\tilde{l}_1\} \cup \ldots \cup \{\tilde{l}_N\}, \end{aligned} \tag{4.42}$$

where $\{l_i + m_i | i = 0, \ldots, 2M+1\}$ and $\{\tilde{l}_j | j = 1, \ldots, N\}$ are solutions to equation (4.33).

Using this solution set in equation (4.36) results in

$$\begin{aligned} Z(k) &= \int_{l_0+m_0^-}^{l_1+m_1^+} u(\phi_1)\mathrm{d}\phi_1 + \int_{l_2+m_2^-}^{l_3+m_3^+} u(\phi_1)\mathrm{d}\phi_2 + \ldots + \int_{l_{2M}+m_{2M}^-}^{l_{2M+1}+m_{2M+1}^+} u(\phi_M)\mathrm{d}\phi_M \\ &\quad + \int_{\tilde{l}_1^-}^{\tilde{l}_1^+} u(\phi_{M+1})\mathrm{d}\phi_{M+1} + \ldots + \int_{\tilde{l}_N^-}^{\tilde{l}_N^+} u(\phi_{M+N})\mathrm{d}\phi_{M+N} \\ &= [U(l_1+m_1^+) - U(l_0+m_0^-)] + [U(l_3+m_3^+) - U(l_2+m_2^-)] + \ldots \\ &\quad + [U(l_{2M+1}+m_{2M+1}^+) - U(l_{2M}+m_{2M}^-)] + [U(\tilde{l}_1^+) - U(\tilde{l}_1^-)] + \ldots \\ &\quad + [U(\tilde{l}_N^+) - U(\tilde{l}_N^-)], \end{aligned} \tag{4.43}$$

where

$$\begin{aligned} &m_i^+ = m_i + \epsilon,\ \epsilon \to 0, && m_i^- = m_i - \epsilon,\ \epsilon \to 0, \\ &\tilde{l}_i^+ = \tilde{l}_i + \epsilon,\ \epsilon \to 0, && \tilde{l}_i^- = \tilde{l}_i - \epsilon,\ \epsilon \to 0, \\ &U(l_i + m_i^+) = \lim_{\epsilon \to 0} U(l_i + m_i + \epsilon), && U(l_i + m_i^-) = \lim_{\epsilon \to 0} U(l_i + m_i - \epsilon), \\ &U(\tilde{l}_i^+) = \lim_{\epsilon \to 0} U(\tilde{l}_i + \epsilon), && U(\tilde{l}_i^-) = \lim_{\epsilon \to 0} U(\tilde{l}_i - \epsilon). \end{aligned} \tag{4.44}$$

Using the assumption that $U(t)$ results from a ZOH, reveals

$$Z(k) = \sum_{n=0}^{M} \{-U[l_{2n} + w(n)] + U(l_{2n+1})\} + \sum_{i=1}^{N} \left\{U(\tilde{l}_i) - U(\tilde{l}_i - 1)\right\}, \quad (4.45)$$

where

$$w(n) = \begin{cases} 0, & m_{2n} > 0 \\ -1, & m_{2n} = 0 \end{cases}. \quad (4.46)$$

According to equation (4.35), all of the $\{l_i | i = 0, \ldots, 2M+1\}$ and $\{\tilde{l}_j | j = 1, \ldots, N\}$ are between $k - l_{max}$ and $k - l_{min}$. Then $Z(k)$ in the form of equation (4.45) can be expressed into the form of equation (4.38).

For every left bound of a subset in the solution (4.42),

$$f_k(l_{2n} + m_{2n}) = 0 \Rightarrow f_k(l_{2n} + 1) \leq 0 \Rightarrow F_k(l_{2n} + 1) = 1. \quad (4.47)$$

If $m_{2n} = 0$,

$$\begin{aligned} f_k(l_{2n}) = 0 &\Rightarrow F_k(l_{2n}) = 1 \\ &\Rightarrow A_{k-l_{2n}}(k) = 0 = F_k(l_{2n}) - F_k(l_{2n} + 1), \quad (4.48) \\ f_k(l_{2n} - 1) > 0 &\Rightarrow F_k(l_{2n} - 1) = 0 \\ &\Rightarrow A_{k-(l_{2n}-1)}(k) = -1 = F_k(l_{2n} - 1) - F_k[(l_{2n} - 1) + 1]. \quad (4.49) \end{aligned}$$

If $m_{2n} > 0$,

$$\begin{aligned} f_k(l_{2n}) > 0 &\Rightarrow F_k(l_{2n}) = 0 \\ &\Rightarrow A_{k-l_{2n}}(k) = -1 = F_k(l_{2n}) - F_k(l_{2n} + 1). \quad (4.50) \end{aligned}$$

For every right bound of a subset in the solution (4.42),

$$f_k(l_{2n+1} + m_{2n+1}) = 0 \Rightarrow f_k(l_{2n+1} + 1) > 0 \Rightarrow F_k(l_{2n+1} + 1) = 0, \quad (4.51)$$

$$\begin{aligned} f_k(l_{2n+1}) \leq 0 \quad &\Rightarrow \quad F_k(l_{2n+1}) = 1 \\ &\Rightarrow \quad A_{k-l_{2n+1}}(k) = 1 = F_k(l_{2n+1}) - F_k(l_{2n+1}+1). \end{aligned} \tag{4.52}$$

For every isolated value in the solution (4.42),

$$\begin{aligned} f_k(\tilde{l}_i - 1) > 0 \quad &\Rightarrow \quad F_k(\tilde{l}_i - 1) = 0, \\ f_k(\tilde{l}_i) = 0 \quad &\Rightarrow \quad F_k(\tilde{l}_i) = 1, \\ f_k(\tilde{l}_i + 1) > 0 \quad &\Rightarrow \quad F_k(\tilde{l}_i + 1) = 0, \\ A_{k-\tilde{l}_i}(k) = 1 \quad &= \quad F_k(\tilde{l}_i) - F_k(\tilde{l}_i + 1), \qquad (4.53) \\ A_{k-(\tilde{l}_i-1)}(k) = -1 \quad &= \quad F_k(\tilde{l}_i - 1) - F_k(\tilde{l}_i). \qquad (4.54) \end{aligned}$$

Equation (4.48) through (4.54) result in

$$A_i(k) = F_k(k-i) - F_k(k-i+1).$$

■

Lemma 4

$$F_k(k+1-l_{min}) = 0, \tag{4.55}$$

$$F_k(k-1-l_{max}) = 1, \tag{4.56}$$

$$F_{k-1}(k-1-l_{max}) = 1. \tag{4.57}$$

Proof: From equation (4.30), $\tau_d(k) \geq l_{min}$, then

$$\begin{aligned} & (k+1-l_{min}) + \tau_d(k+1-l_{min}) \geq k+1 > k \\ \Rightarrow \quad & f_k(k+1-l_{min}) > 0 \\ \Rightarrow \quad & F_k(k+1-l_{min}) = 0. \end{aligned}$$

As $\tau_d(k) \leq l_{max}$,

$$\begin{aligned} & k-1-l_{max}+\tau_d(k-1-l_{max}) \leq k-1 < k \\ \Rightarrow \quad & f_k(k-1-l_{max}) < 0 \\ \Rightarrow \quad & F_k(k-1-l_{max}) = 1. \end{aligned}$$

Also

$$\begin{aligned} & k-1-l_{max}+\tau_d(k-1-l_{max}) \leq k-1 \\ \Rightarrow \quad & f_{k-1}(k-1-l_{max}) \leq 0 \\ \Rightarrow \quad & F_{k-1}(k-1-l_{max}) = 1. \end{aligned}$$

■

Lemma 5

$$\sum_{i=l_{min}}^{j} A_i(k) = F_k(k-j), \qquad j > l_{min}. \tag{4.58}$$

Proof:

$$\begin{aligned} \sum_{i=l_{min}}^{j} A_i(k) &= [F_k(k-l_{min}) - F_k(k-l_{min}+1)] \\ &\quad + [F_k(k-l_{min}-1) - F_k(k-l_{min}-1+1)] \\ &\quad + \ldots + [F_k(k-j) - F_k(k-j+1)] \\ &= F_k(k-j) - F_k(k-l_{min}+1) \\ &= F_k(k-j). \end{aligned} \tag{4.59}$$

■

Define the input cumulative flow difference and output cumulative flow difference as follows

$$\Delta Z(k) = Z(k) - Z(k-1), \tag{4.60}$$

$$\Delta U(k) = U(k) - U(k-1). \tag{4.61}$$

Theorem 4 *If the cumulative input flow is the result of a ZOH, the relation between the input cumulative flow difference and output cumulative flow difference is given by*

$$\Delta Z(k) = \sum_{i=l_{min}}^{l_{max}} B_i(k)\Delta U(k-i) \tag{4.62}$$

where,

$$B_i(k) = \begin{cases} 1, & k-1 < (k-i) + \tau_d(k-i) \leq k \\ 0, & otherwise \end{cases} \qquad l_{min} \leq i \leq l_{max}. \tag{4.63}$$

Proof: Assume $U(0) = 0$, then the cumulative input flow can be written as

$$U(k) = \sum_{i=1}^{k} \Delta U(i). \tag{4.64}$$

From equation (4.38)

$$\begin{aligned} Z(k) &= \sum_{i=l_{min}}^{l_{max}+1} A_i(k) \sum_{j=1}^{k-i} \Delta U(j) \\ &= \sum_{i=l_{min}}^{l_{max}+1} A_i(k)\Delta U(k-i) + \sum_{i=l_{min}}^{l_{max}+1} A_i(k) \sum_{j=1}^{k-i-1} \Delta U(j), \\ Z(k-1) &= \sum_{i=l_{min}}^{l_{max}+1} A_i(k-1) \sum_{j=1}^{k-1-i} \Delta U(j), \\ \Delta Z(k) &= \sum_{i=l_{min}}^{l_{max}+1} A_i(k)\Delta U(k-i) + \sum_{i=l_{min}}^{l_{max}+1} [A_i(k) - A_i(k-1)] \sum_{j=1}^{k-1-i} \Delta U(j) \\ &= \sum_{i=l_{min}}^{l_{max}+1} A_i(k)\Delta U(k-i) + \sum_{j=k-l_{max}-2}^{k-1-l_{min}} \sum_{i=l_{min}}^{k-1-j} [A_i(k) - A_i(k-1)]\Delta U(j) \end{aligned}$$

$$+\sum_{i=l_{min}}^{l_{max}+1}[A_i(k)-A_i(k-1)]\sum_{j=1}^{k-l_{max}-3}\Delta U(j)$$

$$=\sum_{i=l_{min}}^{l_{max}+1}A_i(k)\Delta U(k-i)+\sum_{j=k-l_{max}-2}^{k-1-l_{min}}\Delta U(j)\sum_{i=l_{min}}^{k-1-j}[A_i(k)-A_i(k-1)]$$

$$+\sum_{i=l_{min}}^{l_{max}+1}[A_i(k)-A_i(k-1)]U(k-l_{max}-3). \tag{4.65}$$

From equation (4.58) and equation (4.56)

$$\sum_{i=l_{min}}^{l_{max}+1}A_i(k)=F_k(k-l_{max}-1)=1, \tag{4.66}$$

$$\sum_{i=l_{min}}^{l_{max}+1}A_i(k-1)=F_{k-1}(k-1-l_{max}-1)=1. \tag{4.67}$$

Then, equation (4.65) becomes

$$\begin{aligned}\Delta Z(k) &= \sum_{i=l_{min}}^{l_{max}+1}A_i(k)\Delta U(k-i)+\sum_{j=k-l_{max}-2}^{k-1-l_{min}}\Delta U(j)\sum_{i=l_{min}}^{k-1-j}[A_i(k)-A_i(k-1)]\\ &= \sum_{i=l_{min}}^{l_{max}+1}A_i(k)\Delta U(k-i)+\sum_{j=l_{min}+1}^{l_{max}+2}\Delta U(k-j)\sum_{i=l_{min}}^{j-1}[A_i(k)-A_i(k-1)]\\ &= \sum_{i=l_{min}}^{l_{max}+1}[A_i(k)-A_i(k-1)]\Delta U(k-l_{max}-2)\\ &\quad+\sum_{j=l_{min}+1}^{l_{max}+1}\Delta U(k-j)\left[\sum_{i=l_{min}}^{j-1}[A_i(k)-A_i(k-1)]+A_j(k)\right]\\ &\quad+A_{l_{min}}(k)\Delta U(k-l_{min}).\end{aligned} \tag{4.68}$$

Further, equation (4.68) can be written as

$$\Delta Z(k) = \sum_{i=l_{min}}^{l_{max}+2}B_i(k)\Delta U(k-i), \tag{4.69}$$

where,

$$B_{l_{min}}(k) = A_{l_{min}}(k)$$

$$
\begin{aligned}
&= F_k(k-l_{min}) - F_k(k-l_{min}+1) \\
&= F_k(k-l_{min}), \qquad (4.70)\\
B_i(k) &= \sum_{j=l_{min}}^{i} A_j(k) - \sum_{j=l_{min}}^{i-1} A_j(k-1), \\
&= F_k(k-i) - F_{k-1}(k-i), \\
&\quad \text{where, } l_{min}+1 \le i \le l_{max}+1, \qquad (4.71)
\end{aligned}
$$

and

$$
\begin{aligned}
B_{l_{min}+2}(k) &= \sum_{j=l_{min}}^{l_{max}+1} A_j(k) - \sum_{j=l_{min}}^{l_{max}+1} A_j(k-1) \\
&= F_k(k-l_{max}-1) - F_{k-1}(k-1-l_{max}-1) \\
&= 0 \qquad (4.72)
\end{aligned}
$$

From equation (4.30), $\tau_d(k) \ge l_{min}$, which implies that

$$
\begin{aligned}
&(k-l_{min}) + \tau_d(k-l_{min}) \ge k > k-1 \\
\Rightarrow\ & f_k(k-l_{min}) = (k-l_{min}) + \tau_d(k-l_{min}) - k \ge 0 > -1 \\
\Rightarrow\ & F_k(k-l_{min}) = \begin{cases} 1, & -1 < f_k(k-l_{min}) \le 0 \\ 0, & f_k(k-l_{min}) > 0 \end{cases} \\
\Rightarrow\ & B_{l_{min}} = \begin{cases} 1, & k-1 < k-l_{min}+\tau_d(k-l_{min}) \le k \\ 0, & (k-l_{min}) + \tau_d(k-l_{min}) > k \end{cases} \qquad (4.73)
\end{aligned}
$$

Equation (4.71) can be written as

$$
\begin{aligned}
B_i(k) &= F_k(k-i) - F_{k-1}(k-i) \\
&= \begin{cases} 1, & f_k(k-i) \le 0 \quad \text{and} \quad f_{k-1}(k-i) > 0 \\ 0, & f_k(k-i) > 0 \quad \text{or} \quad f_{k-1}(k-i) \le 0 \end{cases}
\end{aligned}
$$

$$= \begin{cases} 1, & k-1 < (k-i) + \tau(k-i) \leq k \\ 0, & \text{otherwise} \end{cases},$$
$$\text{where}, l_{min} + 1 \leq i \leq l_{max} + 1. \tag{4.74}$$

For the term $B_{l_{max}+1}$, using equation (4.56) and equation (4.57), results in

$$B_{l_{max}+1} = F_k(k - l_{max} - 1) - F_{k-1}(k - l_{max} - 1) = 0. \tag{4.75}$$

Combining equation (4.69) and (4.72) through (4.75), equation (4.62) and (4.63) are obtained. ■

Expressing the relation between the input cumulative flow difference and output cumulative flow difference in matrix form results in

$$\begin{bmatrix} \Delta Z(k) \\ \Delta Z(k+1) \\ \Delta Z(k+2) \\ \vdots \\ \Delta Z(k+N) \end{bmatrix} = \begin{bmatrix} B_{l_{max}}(k) & \dots & B_{l_{min}+1}(k) & B_{l_{min}}(k) \\ 0 & \dots & B_{l_{min}+2}(k+1) & B_{l_{min}+1}(k+1) \\ 0 & \dots & B_{l_{min}+3}(k+2) & B_{l_{min}+2}(k+2) \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & \dots & 0 \\ B_{l_{min}}(k+1) & 0 & \dots & 0 \\ B_{l_{min}+1}(k+2) & B_{l_{min}}(k+2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & \dots & B_{l_{min}}(k+N) \end{bmatrix} \begin{bmatrix} \Delta U(k - l_{max}) \\ \vdots \\ \Delta U(k - l_{min} - 1) \\ \Delta U(k - l_{min}) \\ \Delta U(k + 1 - l_{min}) \\ \Delta U(k + 2 - l_{min}) \\ \vdots \\ \Delta U(k + N - l_{min}) \end{bmatrix}. \tag{4.76}$$

2. Zero-order-hold on input flow rate

In this case, the input flow is assumed to be the output of a controller which dictates the flow rate. In this case, the input flow rate is the result of a discrete time signal passing through a ZOH. The expression for the discrete time cumulative input flow in this case is

$$U(l+1) = U(l) + u(l)T, \tag{4.77}$$

$$\Delta U(l) = U(l) - U(l-1) = u(l-1)T, \tag{4.78}$$

$$U(l+m) = (1-m)U(l) + mU(l+1). \tag{4.79}$$

In this case

$$\lim_{\epsilon \to 0} U(l+m+\epsilon) = \lim_{\varepsilon \to 0} U(l+m-\varepsilon). \tag{4.80}$$

Theorem 5 *If the input flow rate is the output of a ZOH, the relation between the input cumulative flow difference and output cumulative flow difference is:*

$$\Delta Z(k) = \sum_{i=l_{min}}^{l_{max}} B_i(k) \Delta U(k-i) \tag{4.81}$$

where, the calculation of $B_i(k)$ is as given in equation 4.82.

$$
B_i(k) = \begin{cases}
0, & f_k(k-i) > 0, & f_k(k-i-1) > 0; \\
 & \text{or } f_{k-1}(k-i) \le 0, & f_{k-1}(k-i-1) \le 0; \\
1, & f_k(k-i) \le 0, & f_{k-1}(k-i) > 0, \\
 & f_k(k-i-1) \le 0, & f_{k-1}(k-i-1) > 0; \\
1 - m_{a_k}, & f_k(k-i) \le 0, & f_k(k-i-1) > 0, \\
 & f_{k-1}(k-i) \le 0, & \\
 & & f_k(k-i-1+m_{a_k}) = 0; \\
m_{a_{k-1}} - m_{a_k}, & & f_k(k-i-1) > 0, \\
 & f_{k-1}(k-i) \le 0, & \\
 & f_k(k-i-1+m_{a_k}) = 0, & f_{k-1}(k-i-1+m_{a_{k-1}}) = 0; \\
m_{a_{k-1}}, & & f_k(k-i-1) \le 0, \\
 & f_{k-1}(k-i) \le 0, & f_{k-1}(k-i-1) > 0, \\
 & & f_{k-1}(k-i-1+m_{a_{k-1}}) = 0; \\
m_{b_k} - m_{b_{k-1}}, & f_k(k-i) > 0, & \\
 & & f_{k-1}(k-i-1) \le 0, \\
 & f_k(k-i-1+m_{b_k}) = 0, & f_{k-1}(k-i-1+m_{b_{k-1}}) = 0; \\
1 - m_{b_{k-1}}, & f_k(k-i) \le 0, & \\
 & f_{k-1}(k-i) > 0, & f_{k-1}(k-i-1) \le 0, \\
 & & f_{k-1}(k-i-1+m_{b_{k-1}}) = 0; \\
m_{b_k}, & f_k(k-i) > 0, & f_k(k-i-1) \le 0 \\
 & & f_{k-1}(k-i-1) > 0, \\
 & f_k(k-i-1+m_{b_k}) = 0; &
\end{cases}
\tag{4.82}
$$

Proof: For this case, equation (4.43) becomes

$$\begin{aligned}
Z(k) &= \{[(1-m_1)U(l_1)+m_1U(l_1+1)]-[(1-m_0)U(l_0)+m_0U(l_0+1)]\} \\
&\quad +\{[(1-m_3)U(l_3)+m_3U(l_3+1)]-[(1-m_2)U(l_2)+m_2U(l_2+1)]\} \\
&\quad +\cdots+\{[(1-m_{2M+1})U(l_{2M+1})+m_{2M+1}U(l_{2M+1}+1)] \\
&\quad -[(1-m_{2M})U(l_{2M})+m_{2M}U(l_{2M}+1)]\} \\
&= \sum_{n=0}^{M}\{[(1-m_{2n+1})U(l_{2n+1})+m_{2n+1}U(l_{2n+1}+1)] \\
&\quad -[(1-m_{2n})U(l_{2n})+m_{2n}U(l_{2n}+1)]\} \\
&= \sum_{n=0}^{M}\{\{U(l_{2n+1})+m_{2n+1}[U(l_{2n+1}+1)-U(l_{2n+1})]\} \\
&\quad -\{U(l_{2n})+m_{2n}[U(l_{2n}+1)-U(l_{2n})]\}\} \\
&= \sum_{n=0}^{M}\{[U(l_{2n+1})+m_{2n+1}\Delta U(l_{2n+1}+1)]-[U(l_{2n})+m_{2n}\Delta U(l_{2n}+1)]\} \\
&= \sum_{n=0}^{M}\{[U(l_{2n+1})-U(l_{2n})]+m_{2n+1}\Delta U(l_{2n+1}+1)-m_{2n}\Delta U(l_{2n}+1)\} \\
&= \sum_{n=0}^{M}\left[\sum_{i=l_{2n}+1}^{l_{2n+1}}\Delta U(i)+m_{2n+1}\Delta U(l_{2n+1}+1)-m_{2n}\Delta U(l_{2n}+1)\right] \\
&= \sum_{n=0}^{M}\left[m_{2n+1}\Delta U(l_{2n+1}+1)+\sum_{i=l_{2n}+2}^{l_{2n+1}}\Delta U(i)+(1-m_{2n})\Delta U(l_{2n}+1)\right] \\
&= (1-m_0)\Delta U(l_0+1)+\sum_{i=l_0+2}^{l_1}\Delta U(i)+m_1\Delta U(l_1+1)+\ldots \\
&\quad +(1-m_{2M})\Delta U(l_{2M}+1)+\sum_{i=l_{2M}+2}^{l_{2M+1}}\Delta U(i) \\
&\quad +m_{2M+1}\Delta U(l_{2M+1}+1). \qquad (4.83)
\end{aligned}$$

Equation (4.83) can be cast in the form of

$$Z(k)=\sum_{i=l_{min}}^{\infty}C_i(k)\Delta U(k-i), \qquad (4.84)$$

where,

$$C_i(k) = \begin{cases} 0, & f_k(k-i) > 0, \quad f_k(k-i-1) > 0; \\ 1 - m_a, & f_k(k-i) \leq 0, \quad f_k(k-i-1) > 0, \quad f_k(k-i-1+m_a) = 0; \\ 1, & f_k(k-i) \leq 0, \quad f_k(k-i-1) \leq 0; \\ m_b, & f_k(k-i) > 0, \quad f_k(k-i-1) \leq 0, \quad f_k(k-i-1+m_b) = 0. \end{cases} \tag{4.85}$$

The difference of cumulative output flow is calculated as

$$\begin{aligned} \Delta Z(k) &= Z(k) - Z(k-1) \\ &= \sum_{i=l_{min}}^{\infty} C_i(k)\Delta U(k-i) - \sum_{i=l_{min}}^{\infty} C_i(k-1)\Delta U(k-i-1) \\ &= \sum_{i=l_{min}}^{\infty} C_i(k)\Delta U(k-i) - \sum_{j=l_{min}+1}^{\infty} C_{j-1}(k-1)\Delta U(k-j) \\ &= C_{l_{min}}\Delta U(k-l_{min}) + \sum_{i=l_{min}+1}^{\infty} [C_i(k) - C_{i-1}(k-1)]\Delta U(k-i). \end{aligned} \tag{4.86}$$

Equation (4.86) can be written in the following form,

$$\Delta Z(k) = \sum_{i=l_{min}}^{\infty} B_i(k)\Delta U(k-i), \tag{4.87}$$

where

$$B_i(k) = [C_i(k) - C_{i-1}(k-1)], \qquad \text{for } i = l_{min}+1, \ldots, \infty, \tag{4.88}$$

$$B_{l_{min}}(k) = C_{l_{min}}(k). \tag{4.89}$$

For $B_{l_{max}+p}(k)$, where $p \geq 1$,

$$B_{l_{max}+p}(k) = C_{l_{max}+p}(k) - C_{l_{max}+p-1}(k-1), \tag{4.90}$$

$$
\begin{aligned}
\because f_k(k-l_{max}-p) &= (k-l_{max}-p)+\tau_d(k-l_{max}-p)-k \\
&\leq k-l_{max}-p+l_{max}-k \\
&= -p<0, \\
\because f_k(k-l_{max}-p-1) &= (k-l_{max}-p-1)+\tau_d(k-l_{max}-p-1)-k \\
&\leq (k-l_{max}-p-1)+l_{max}-k \\
&= -p-1<0, \\
\therefore C_{l_{max}+p}(k) &= 1, \qquad (4.91)
\end{aligned}
$$

$$
\begin{aligned}
\because f_{k-1}(k-l_{max}-p) &= (k-l_{max}-p)+\tau_d(k-l_{max}-p)-(k-1) \\
&\leq k-l_{max}-p+l_{min}-k+1 \\
&= -p+1\leq 0, \\
\because f_{k-1}(k-l_{max}-p-1) &= (k-l_{max}-p-1)+\tau_d(k-l_{max}-p-1)-(k-1) \\
&\leq (k-l_{max}-p-1)+l_{max}-k+1 \\
&= -p<0, \\
\therefore C_{l_{max}+p-1}(k-1) &= 1. \qquad (4.92)
\end{aligned}
$$

From equation (4.90) through (4.92), it can be obtained that

$$B_{l_{max}+p}(k)=0. \qquad (4.93)$$

For $B_{l_{min}}(k)$,

$$
\begin{aligned}
\because f_k(k-l_{min}) &= (k-l_{min})+\tau_d(k-l_{min})-k \\
&\geq (k-l_{min})+l_{min}-k=0, \\
\therefore B_{l_{min}}(k) &= C_{l_{min}}(k)
\end{aligned}
$$

$$= \begin{cases} 0, & f_k(k-l_{min}-1) > 0; \\ 1, & f_k(k-l_{min}-1) \le 0, \quad f_k(k-l_{min}) = 0; \\ m_b, & f_k(k-l_{min}-1) < 0, \quad f_k(k-l_{min}) > 0, \\ & \qquad\qquad\qquad\quad f_k(k-l_{min}-1+m_b) = 0. \end{cases} \tag{4.94}$$

For $B_i(k)$, where $l_{min} < i \le l_{max}$,

$$B_i(k) = C_i(k) - C_{i-1}(k-1) \tag{4.95}$$

$$C_i(k) = \begin{cases} 0, & f_k(k-i) > 0, \quad f_k(k-i-1) > 0; \\ 1-m_{a_k}, & f_k(k-i) \le 0, \quad f_k(k-i-1) > 0, \\ & \qquad\qquad\quad f_k(k-i-1+m_{a_k}) = 0; \\ 1, & f_k(k-i) \le 0, \quad f_k(k-i-1) \le 0; \\ m_{b_k}, & f_k(k-i) > 0, \quad f_k(k-i-1) \le 0, \\ & \qquad\qquad\quad f_k(k-i-1+m_{b_k}) = 0.; \end{cases} \tag{4.96}$$

and

$$C_{i-1}(k-1) = \begin{cases} 0, & f_{k-1}(k-i) > 0, \quad f_{k-1}(k-i-1) > 0; \\ 1-m_{a_{k-1}}, & f_{k-1}(k-i) \le 0, \quad f_{k-1}(k-i-1) > 0, \\ & \qquad\qquad\qquad f_{k-1}(k-i-1+m_{a_{k-1}}) = 0; \\ 1, & f_{k-1}(k-i) \le 0, \quad f_{k-1}(k-i-1) \le 0; \\ m_{b_{k-1}}, & f_{k-1}(k-i) > 0, \quad f_{k-1}(k-i-1) \le 0, \\ & \qquad\qquad\qquad f_{k-1}(k-i-1+m_{b_{k-1}}) = 0. \end{cases} \tag{4.97}$$

Combine equation (4.95) through (4.97), equation (4.82) is obtained. ■

E. Source Buffering Based Predictive Control

For improving the QoS of a single real-time multimedia flow, a source buffering based predictive controller using the derived fluid model is attempted. A block diagram of the system with a predictive controller using source buffering is presented in Fig. 10.

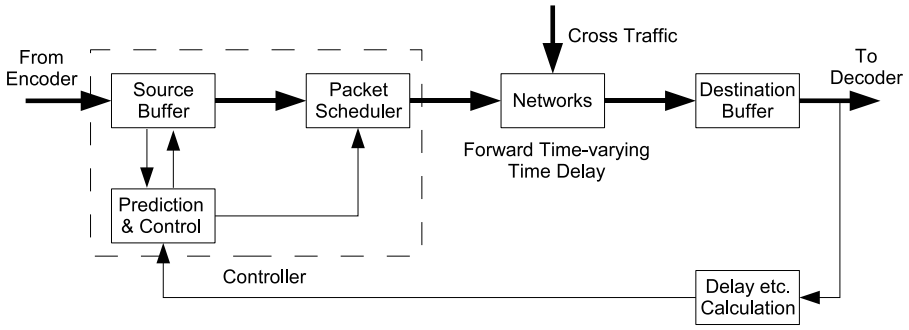

Fig. 10. Block diagram of the system with a predictive controller using source buffering.

The concept of source buffering has been used by Konstantinou in his research with a white box single router fluid model of a network to improve the QoS of real-time multimedia applications [116] and by Khariwal in his research with a black box ARX model of the flow accumulation [130]. Comparing this concept with the commonly used receiver side buffering, source buffering enables the possibility of rescheduling the time when packets enter the network. Konstantinou suggested that through source buffering it is possible to reduce the loss rates with some increase in the end-to-end delay of packets [116]. Khariwal demonstrated that, together with a destination buffer, source buffering can reduce the disruption in the playback of a multimedia stream and the initial buffering time, when the lose rate of the networks is between 2% to 8% [130].

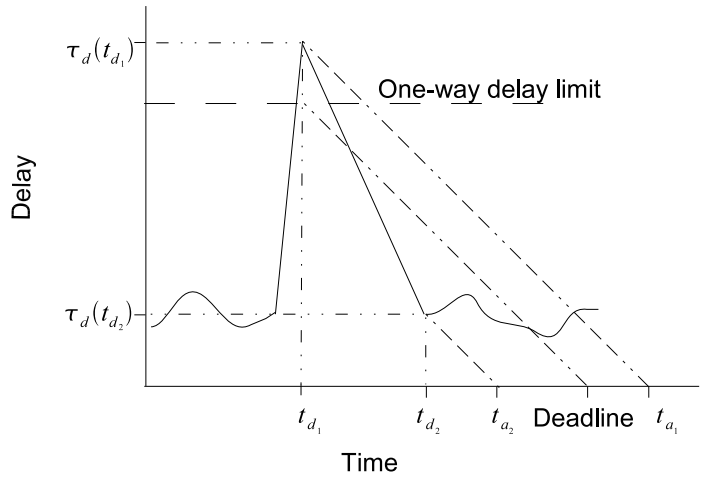

Fig. 11. An example of source buffering in case of flow reversal.

1. The case of network flow reversal

Under the assumption that a single flow of interest is relatively small compared to the other flows in a network and that the delay experienced by that single flow is independent of the flow rate of that single flow, if there is flow reversal in the system, then based on equation (4.3), for some period of time the time derivative of the delay is less than -1, i.e. $\dot{\tau}_d(t) \leq -1$. This is illustrated in Fig. 11 with an example. If a packet was sent from the source at time t_{d_1}, it would experience the delay $\tau_d(t_{d_1})$, and it would arrive at the destination at time t_{a_1}. However, if a source buffer was implemented and the packet was held back in the source buffer until t_{d_2}, it would experience the delay $\tau_d(t_{d_2})$, and arrive at time t_{a_2}. Assuming the period between t_{d_1} and t_{d_2} to be a period when the the derivative of the delay is less than -1, then the average derivative of delay $\bar{\dot{\tau}}_d$ is also less than -1, thus

$$\frac{\tau_d(t_{d_2}) - \tau_d(t_{d_1})}{t_{d_2} - t_{d_1}} = \bar{\dot{\tau}}_d < -1$$
$$\Rightarrow \quad \tau_d(t_{d_2}) - \tau_d(t_{d_1}) < -t_{d_2} + t_{d_1}$$

$$\Rightarrow \quad \tau_d(t_{d_2}) + t_{d_2} < \tau_d(t_{d_1}) + t_{d_1}$$
$$\Rightarrow \quad t_{a_2} < t_{a_1}. \tag{4.98}$$

Holding back the packet in the source buffer until t_{d_2} would result in the packet arrival at the destination at t_{a_2}, earlier than if the packet is introduced in the network at t_{d_1}. In this case, even if the delay at time t_{d_1} is more than the one-way delay limit for interactivity, an a packet sent at that time would eventually miss its playback deadline, it would still be possible to meet the playback deadline should the packet be held back until a later time. So, if there is flow reversal in the system, it is possible to improve the performance of real-time multimedia applications through source buffering.

2. The case of no network flow reversal

under the assumption that a single flow of interest is relatively small compared to the other flows in a network and that the delay experienced by that single flow is independent of the flow rate of that single flow, if there is no flow reversal in the system, then equation (4.3) holds, and the the derivatives of the delay is always more than -1, i.e. $\dot{\tau}_d(t) > -1$. The average derivative of delay $\bar{\dot{\tau}}_d$ of any time period is more than -1, thus

$$\frac{\tau_d(t_{d_2}) - \tau_d(t_{d_1})}{t_{d_2} - t_{d_1}} = \bar{\dot{\tau}}_d > -1$$
$$\Rightarrow \quad \tau_d(t_{d_2}) - \tau_d(t_{d_1}) > -t_{d_2} + t_{d_1}$$
$$\Rightarrow \quad \tau_d(t_{d_2}) + t_{d_2} > \tau_d(t_{d_1}) + t_{d_1}$$
$$\Rightarrow \quad t_{a_2} > t_{a_1}. \tag{4.99}$$

This situation is demonstrated in Fig. 12. In this case, holding back a packet will make it always arrive later than not holding it back. In this case, if a packet is

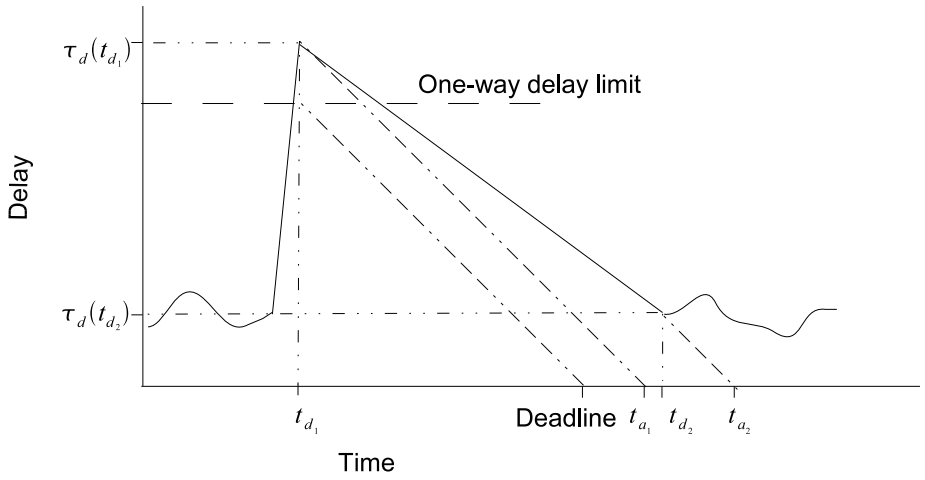

Fig. 12. An example of source buffering in case of no flow reversal.

sent at time t_{d_1} it would experience a delay more than the one-way delay limit for interactivity and it would miss its playback deadline. Holding it back would not help it meeting the deadline either. So, if there is no flow reversal in the system, source buffering is not an effective method for improve the performance of real-time multimedia applications.

F. Literature Review on Flow Reversal

Section E shows that only during periods with network flow reversal will source buffering based predictive control be effective for improving the quality of interactive multimedia applications. In order for this control method to have significant improvement for these applications, there should be sufficient flow reversal in network. The question of what is sufficient flow reversal i an open-ended question. However, one can ascertain that current measured levels of flow reversal in public IP networks.

Bennett and Partridgereported a high rate of reordering (90% of the sessions) in

the Internet, using ICMP probing packets and MAE-East exchange [18]. Most of the reordering was contributed to the parallelism in the Internet. In 2000, the Internet End-to-end Performance Monitoring (IEPM) group at Stanford Linear Accelerator Center (SLAC) reported that about 25% of the 250 hosts monitored with ICMP probing packets exhibited reordering [19]. The average reordering rate they reported is 16%. They also reported that "reordering is high to the developing world and to commercial Internet sites (.com and .net)". From these reports it seems that there is enough reordering in the Internet to make source buffering based predictive control an effective method to improve the quality of interactive multimedia applications.

Yet in 2002 Jaiswal et al. reported only 0.02% to 0.5% reordering of packets, when monitoring the Sprint IP backbone using TCP packets [20]. In 2004 Gharai et al. reported reordering rates from 0.01% to 1.65%, when performing measurements within the US, using UDP packets [21]. These reports suggested a much lower rate of packet reordering in the Internet compared to the previous reports.

Bellardo and Savage [22] reported that the reordering rate is related to the inter-departure time of the packets. They reported that when sending back-to-back packets there is 10% reordering, which drops to less than 2% with 50 microseconds of inter-departure time, and drops to almost 0% with 250 microseconds inter-departure time. In many multimedia applications, the interdeparture time used is about 20 millisecond (or $20,000$ microseconds), compared to the above observation, it is expected that there is not much packet reordering in multimedia applications. Gharai et al. [21] reported a similar trend, with the reordering rate dropping when the inter-departure time of packets changed from 20 microseconds to 40 microseconds. In Bennett's report [18] and IEPM's report [19], the packets were sent back-to-back with very little inter-departure time, while in Jaiswal's report [20] and Gharai's report [21] the packets were with much larger interdeparture time. This might be one reason for the

difference in the observed reordering rates.

The above literature suggests that only when packets are sent very fast, on the order of tens of microseconds, will packet reordering (or flow reversal) be a significant phenomenon. It might be present in TCP applications when transferring large bulk of files. But for real-time multimedia applications, such as VoIP, the normal inter-departure time is on the order of tens of milliseconds, in which case there will be very little packet reordering. Thus source buffering based predictive control is not a very effective method for improving the QoS of real-time multimedia applications under current Internet traffic conditions.

G. Regarding Losses

If the no losses assumption is not true, the system is no longer a conservative system, the fluid model derived in the previous sections need to be modified. Konstantinou suggested that it is possible to reduce the loss rates of a single flow by source buffering, with a trade-off of some increase in the end-to-end delay [116]. If a packet would get lost if it was sent at time t_{d_1}, and would not get lost if it was sent at a later time t_{d_2}, then holding it back until t_{d_2} might reduce the losses. So source buffering would improve the QoS of multimedia applications in this case.

In the case of no flow reversal in the system, lost packets can be detected when the subsequent packets are starting to arrive at the destination. Yet in the case when there is flow reversal in the system, following a packet arrival at the destination, it is impossible to tell whether the earlier packets that have yet to arrive are lost or just being delayed. In this case the losses are undetectable. Thus a loss signal cannot be measured, predicted, and used for control decision. The only way to detect losses might be to implement a time-out mechanism such that if a packet has not arrived

within a given time limit it is assumed to be lost.

Even if packet losses are detected, it is still very difficult to predict them accurately enough for the source buffering control to improve the quality of real-time multimedia applications. There is a lot of research on modeling the loss process [127], on predicting the long term average loss rate [128], and on predicting the likelihood of having losses during a time interval [129]. But there are few publications on the prediction of losses at the packet level. Against the most common belief that an increase in delay is a good indicator for packet loss prediction, Marin et al. showed that "the level of correlation between a increase in RTT and packet loss is not strong enough to allow a TCP/Sender to reliably improve throughput" [16]. Thus it is very difficult to utilize the loss signal in a predictive controller using source buffering.

H. Chapter Summary

In this chapter the possibility of improving the quality of real-time multimedia applications using a fluid model based single flow control, i.e. the source buffering based predictive control, has been investigated.

First the assumptions behind the approach used in this chapter are discussed. A continuous fluid model of a signal flow transported over a network system without flow reversal has been introduced. The case of a network with flow reversal is also derived. Then the fluid model is discretized based on different assumptions on the input flow. The relation between the difference of input cumulative input flow and the difference of cumulative output flow is derived for each case. Next, source buffering based predictive control is introduced. It is demonstrated that source control is effective for improving the quality of interactive multimedia applications only during periods when there is flow reversal in the network, under the assumption that a

single flow of interest is relatively small compared to the other flows in a network and that the delay experienced by that single flow is independent of the flow rate of that single flow. Finally, literature review of research related to flow reversal reveals that for applications of interest to this research, today's Internet does not show sufficient packet reordering for the the source buffering based predictive control to be an effective method for QoS improvement.

CHAPTER V

SINGLE FLOW CONTROL THROUGH PREDICTIVE PATH SWITCHING

In this chapter, firstly, the problem statement is given. The idea of using predictive path switching control to improve VoIP QoS is introduced, and the idea of using dynamic system models for the predictor development is proposed. The general assumption used in this study is discussed, and several concepts used in predictive path switching control are introduced. Secondly, the required prediction horizon of a predictor developed for predictive path switching control is discussed. Then the emulation study method of predictive path switching control using artificially generated traffic profiles is introduced. Finally, emulation studies of the impact of traffic delay signal frequency content and path comprehensive loss rate on the predictive path switching control method are performed using artificially generated traffic profiles.

A. Introduction to Predictive Path Switching Control

The past decade has seen an increasing number of real-time multimedia applications, such as VoIP applications, running on the Internet. But the lack of quality guarantee in best-effort networks has hampered their widespread popularity. So improvements in the QoS for VoIP applications in best-effort networks has raised a lot of research interest.

The increasing number of Internet Service Providers (ISPs) has made it possible to have *path-diversity* through having multiple Internet connections from different ISPs. The same can also be achieved through an overlay network. If there is path-diversity available, then it is possible to find a better path for a given time. The VoIP packets can be transmitted over the better path at each time instant, potentially improving VoIP QoS. The method of dynamically switching among the available paths

to send the application packets is called *path switching* [6].

Recently, Tao et al. has done some work with this method [50, 51, 102, 119]. Their research [50] shows that by implementing path switching, the average delay experienced by the application packets cannot be improved much. But the average packet loss rate experience by the application packets can be reduced.

In these studies, the authors were looking at time scales from minutes [50] to tens of seconds [102]. In the current research, real-time multimedia applications, such as VoIP, are considered where time scales of the order of one second can impact the transmission quality. Therefore, it might be better to perform control on time scales below one second.

By comparing the results of Markov model based predictors and simple predictors, Tao and Guerin found that "a simple predictor, in most cases, performs as well as the complex ones" [119]. But the authors did not consider many other types of models. In the current research, some linear dynamic system models, such as the autoregressive (AR) model, and some nonlinear dynamic system models are considered.

In the research by Tao et al., the path switching was done based on MOS calculated from the E-model. The path under investigation had packet loss rate below 2% [102], and MOS above 3. In their test-bed data case, they have improved the MOS from 3.7 to 3.9. But MOS of 3.7 is already toll quality, and there is not much need to improve it. In the current study, bad network paths, which will normally give MOS below 3 for VoIP applications, are considered for possible improvements.

Problem Statement: *The thrust of this research is to study the ability of dynamic path switching in improving the QoS of VoIP applications, for paths which normally have sufficient congestion to provide bad voice quality. The effect of using a predictive control scheme with this method is investigated. Dynamic system models are considered for making the predictions.*

B. General Assumption and Concepts in Predictive Path Switching Control

1. General assumption

In the current study, it is assumed that the VoIP flow of interest is a relatively small amount of the flow compared to all the other flows in the network. Thus the network condition is dominated by the cross flow. Whatever this VoIP flow does has insignificant impact on the network's condition. Particularly, the flow rate of this flow has minimal or no impact on the delay and loss rate it will experience when passing through over a given path.

Under this assumption, it is possible to send probing packets through a path, measure their delays and losses, and claim that these will be the same delays and losses for the VoIP packets, should they be sent through the same path at the same time. The delays and losses collected from a probing experiment of the path are called the *trace-file* of that path.

If this assumption is not true, then the VoIP flow of interest is a relatively significant amount of the flow compared to all the other flows in the network. The flow rate of this VoIP flow will have impact on the delay and loss rate it will experience when passing through a given path. Then the delays and losses measured by the probing packets will not be the same delays and losses for the VoIP packets, should they be sent through the same path at the same time. Whether predictive path switching control can improve the QoS of real-time multimedia applications in this case and if yes then how to do it are still open questions.

2. One-way delay limit

The ITU-T G.114 [148] suggests that one-way delay is to be kept below 150 ms, so that most applications would not be significantly affected. It also recommends not to

exceed a one-way delay of 400 ms as the upper limit. In industry, the one-way delay limit for VoIP calls is normally 200 ms to 250 ms [149].

If VoIP packets are sent at 20 ms interval, the collection of voice samples will take 20 ms. The Speex codec used in this research will take an encoding time of 5 ms. The sending and receiving take about 3 ms, the decoding takes about 1 ms. So a network delay plus jitter buffering time of 150 ms is a reasonable choice to have for a delay budget of below 200 ms.

If VoIP packets are sent at 100 ms interval, the collection of voice samples will take 100 ms. The Speex codec will take 7 ms for encoding. The sending and receiving take about 3 ms. The decoding takes about 1 ms. A network delay plus jitter buffering time of 150 ms will give a delay budget of slightly over 260 ms, which is still acceptable.

So in this study, 150 ms is set to be the one-way delay limit for the network delay. VoIP packets with network delay lower than this will be buffered to this time limit before playback, while packets with network delay over this will be dropped and considered lost. A packet exceeds the one-way delay limit is called an over-delayed packet.

3. Comprehensive loss rate

The voice quality of a VoIP application is affected by both the packet loss (or drop) rate and the packet delay it has experienced. Both should be taken into account for judging the quality of a path. Any packet that exceeds the one-way delay limit is as bad as being lost [78]. So the packet loss rate for VoIP applications should include both the packet loss (or drop) rate in the network path, and the rate of the VoIP packets being over-delayed. In this study, the ratio of the number of packets dropped in a network path plus the number of over-delayed packets in that path to the number of total packets sent through it is called the comprehensive loss rate (CLR) of that

path. When CLR is calculated for a very small time interval it is different from calculated it for a long time interval. If calculated for a very small time interval, it is like a instantaneous signal. If calculated for at long time interval it gives an average value. The CLR for time interval $[t_1, t_2)$ is given by

$$CLR[t_1, t_2] = \frac{N_{loss}[t_1, t_2] + N_{overdelay}[t_1, t_2]}{N_{total}[t_1, t_2]} \times 100\% \tag{5.1}$$

where, $N_{loss}[t_1, t_2]$ is the number of dropped packets in this interval, $N_{overdelay}[t_1, t_2]$ is the number of over-delayed packets in this interval, $N_{total}[t_1, t_2]$ is the total number of packets in this interval. If $t_2 = t_1 + \delta$, as $\delta \rightarrow 0$, then the CLR approaches an instantaneous value. But if $\delta >> 0$ then a more average value is obtained.

4. The information feedback delay limit

When probing the available paths using probing packets from the sender side, before the measurement can be obtained at the sender side, firstly, the probing packet has to be transmitted forward to reach the receiver side, then it has to be responded back by the receiver, and finally it has to be transmitted backward to the sender. So at any given time, at the sender side only the information one round trip time (RTT) ago is known.

As for VoIP applications, their packets have to meet the one-way delay limit in order to preserve the interactivity [148]. In this study, this one-way delay limit is set to 150 ms. So the one-way delay experienced by a probing packet is no more than 150 ms, or it is considered as a measured lost. This 150 ms forward one-way delay limit is designated as τ_f. The path is probed every 100 ms from the sender side. The receiver side is also probing the paths every 100 ms. The response to a sender side probing packet is piggybacked to a probing packets sent from the receiver side. It will take at most 100 ms for the response of a correctly received sender side probing

packet to be sent to the sender. Designate this maximum response time as τ_r. This response should also be received at the sender side within 150 ms or is considered lost. This backward one-way delay limit is designated as τ_b. This gives an upper bound of

$$\tau = \tau_f + \tau_r + \tau_b = 150 + 100 + 150 = 400\,(\text{ms}), \tag{5.2}$$

for a sender side probing packet to be transmitted forward and then feedback. If the response to a probing packet has not been received within 400 ms after its sending time, it is considered lost.

This upper bound (τ) of 400 ms is called an *information feedback delay limit*. For any probing packet that was sent more than 400 ms ago, either its response is correctly received and the measurement is successfully obtained, or it is considered lost. Thus, all the past condition of the probed path more than 400 ms ago is known. Fig. 13 gives an illustration of the information feedback delay limit.

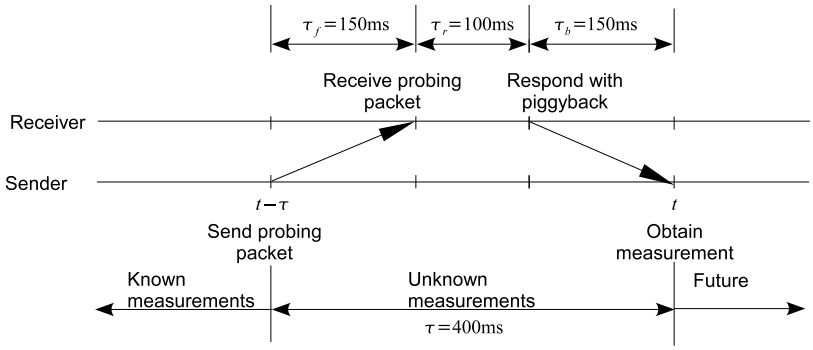

Fig. 13. Information feedback delay limit.

This network's condition measuring method has taken into account not only the delays and losses of the forward path, which is the primary information of interest, but also the impact of the delays and losses in the backward path. In this study, it is assumed that there is no losses in the backward path, and the backward delay

is always below the 150 ms one-way delay limit. Even if this assumption is true, it would still make sense to take the condition of the backward path into consideration and avoid a path that has a lot of losses and large delays in the backward direction as well. Thus this 400 ms information feedback delay limit method is still OK.

C. Emulation Study Method

In this section, an emulation study of the predictive path switching method is performed. The intent is to explore whether path switching control can improve VoIP quality in the presence of path-diversity and to explore if dynamic system model based predictors perform better than simple predictors.

1. Generation of trace-files for traffic delays and losses

In this emulation study, the trace-files of two paths are generated from a combination of Pareto distribution and Gaussian distribution as in NistNet [111], a networks emulation tool. The Gaussian distribution is also called the normal distribution. Its probability density function (pdf) is given by [150]

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{5.3}$$

where, μ is the mean and σ is the standard deviation. The Pareto distribution is a power law probability distribution [151]. The probability that a random variable X, drawn from the Pareto distribution, is greater than some number x is given by

$$Pr(X > x) = \left(\frac{x}{x_m}\right)^{-k}, \tag{5.4}$$

for all $x \geq x_m$, where $x_m > 0$ is the minimum possible value of X, and k is a positive parameter. Its probability density function is

$$f(x; k, x_m) = k\frac{x_m^k}{x^{k+1}} \text{ for } x \geq x_m. \tag{5.5}$$

The software used for generating the delays and losses is obtained from the Nist-Net source code. A delay value is generated as a combination of one forth of a value from the Gaussian distribution and a three forth of a value from the Pareto distribution, "a combination which seems to match experimentally observed distributions reasonably well, but is computationally easy to handle" [111]. A minimum delay is imposed on the generated trace-files by replacing the generated delay values which are less than the required minimum value with the minimum delay value. The losses are randomly generated to give a desired loss rate.

In this emulation study, the packet losses in the trace-files are randomly generated to give a packet loss rate of 1%. By adjusting the variation of the delay signals the desired average CLR of each emulation case is obtained. Then the generated delay signals are passed through low pass filters of different cut-off frequencies, to simulate the scenarios where the delay signals have different frequency contents.

In this emulation study, it is assumed that the VoIP packets are encoded with the Speex codec, the inter-departure time for the VoIP packets is 100 ms, and there are five frames per packet. Thus the trace-files are also generated with a sampling interval of 100 ms (a sampling rate of 10 Hz).

2. Predictive path switching controller

Assuming that there are two paths available for selection. Because in this study, the inter-departure time for the VoIP packets is 100 ms, so the two possible paths are also probed every 100 ms. The probing packets from the source host are piggybacked

by the probing packets from the destination host, which are also sent every 100 ms.

Because of the information feedback delay limit of 400 ms, the network condition in the past 400 ms is unknown to the controller. Only the network condition more than 400 ms ago are known to the controller. If it is decided that the path switching decision is made and implemented also every 400 ms, then to make a control decision for the next 400 ms interval, at least two-step-ahead predictions of the network condition in the two paths are needed.

Two signals are used to reflect the network condition in this emulation study. One is the comprehensive loss rate (CLR) of every 400 ms interval of each of the two paths. The other is the average delay over every 400 ms of each of the two paths. Two types of predictors are developed for each of the two types of signals. One is a simple predictor (SP), and the other is an autoregressive (AR) predictor.

Assume that the switching is to be done between paths A and B. The two-step-ahead simple predictors for Path A and Path B are given by

$$\begin{aligned} \hat{y}_{A,SP}(k) &= y_A(k-2), \\ \hat{y}_{B,SP}(k) &= y_B(k-2), \end{aligned} \tag{5.6}$$

where k denotes the current time step, $\hat{y}_{A,SP}(k)$ and $\hat{y}_{B,SP}(k)$ are the predicted signals at the current time step for Path A and Path B, and $y_A(k-2)$ and $y_B(k-2)$ are the latest available measurements of these two signals.

The two-step-ahead AR predictors for Path A and Path B are given by

$$\begin{aligned} \hat{y}_{A,AR}(k) &= a_{1,A}y_A(k-2) + a_{2,A}y_A(k-3) + \ldots + a_{n_{a,A},A}y_A(k-n_{a,A}-1) \\ \hat{y}_{B,AR}(k) &= a_{2,B}y_B(k-2) + a_{2,B}y_B(k-3) + \ldots + a_{n_{a,B},B}y_B(k-n_{a,B}-1) \end{aligned}, \tag{5.7}$$

where, k denotes the current time step, $\hat{y}_{A,AR}(k)$ and $\hat{y}_{B,AR}(k)$ are the predicted signals at the current time step for Path A and Path B, and $\{y_A(i)|i = k-2, \ldots, k-n_{a,A}-1\}$

and $\{y_B(i)|i = k-2, \ldots, k-n_{a,B}-1\}$ are the latest available measurements of these two signals, $\{a_{i,A}|i = 1, \ldots, n_{a,A}\}$ and $\{a_{i,B}|i = 1, \ldots, n_{a,B}\}$ are the predictor coefficients of the Path A predictor and the Path B predictor, respectively, $n_{a,A}$ and $n_{a,B}$ are the orders of the two predictors, respectively.

The AR predictors are developed using half of the trace-file of a path as the training set to find the coefficients for a given order through the least squares method, while using the rest half of that trace-file as the testing set to find the best predictor of the that path. Mean square error (MSE)

$$MSE = \frac{\sum_{k=1}^{N} \hat{y}_{AR}(k) - y(k)}{\sum_{k=1}^{N} y^2(k)}, \tag{5.8}$$

where, $\{\hat{y}_{AR}(k)|k = 1, \ldots, N\}$ are the predictions of the signal and $\{y(k)|k = 1, \ldots, N\}$ are the measurements of the signal, is used for finding the best predictor. The predictor which gives the minimum MSE on the test set is chosen as the best predictor.

For predictive path switching control, at each time step k, once the prediction results $\hat{y}_A(k)$ and $\hat{y}_B(k)$ are obtained, they are compared, and the path with the better quality, i.e. the smaller CLR or delay value, is chosen for the VoIP packets to be transmitted for the next 400 ms. If the quality of the two paths is the same, then the path chosen in the previous time step is chosen.

In no switching and transmitting over Path A case, all the VoIP packets are transmitted over Path A and experience all the delays and losses of Path A. In no switching and transmitting over Path B case, all the VoIP packets are transmitted over Path B, and experience all the delays and losses of Path B. These results show the original VoIP QoS over each path when there is no control.

For ideal case path switching control, it is assumed that the real signals $y_A(k)$ and $y_B(k)$ are known ahead of time, and are used directly in the path switching control. This gives an upper limit of the possible QoS improvement that can be achieved by

path switching.

3. Voice quality evaluation

With the aforementioned predictive path switching controller, the new CLRs and delays experienced by the VoIP packets can be determined. These loss rates and delays are to be used in the E-model of the Speex codec. The resulting R-values are mapped into MOSs to compare the results.

As the delay used in the E-model should be the mouth-to-ear delay, a sample collection delay of 100 ms, encoding delay of 7 ms, sending and receiving delay of 3 ms, and decoding delay of 1 ms should also be added to the network delay experienced by the VoIP packets.

The E-model MOSs of the resulting voice are calculated for each of the following VoIP packets transmission methods: no switching and transmitting over Path A (A), no switching and transmitting over Path B (B), ideal case path switching control ($ideal$), predictive path switching control using CLR signals and simple predictors (SP_{CLR}), predictive path switching control using delay signals and simple predictors (SP_{delay}), predictive path switching control using CLR signals and AR predictors (AR_{CLR}), and predictive path switching control using delay signals and AR predictors (AR_{delay}).

D. Emulation Study Results

1. Impact of traffic delay signal frequency content

a. The case of 5% comprehensive loss rate

In this set of emulations, the variances of the Pareto distribution and Gaussian distribution, which are combined to generate the delay signals, are adjusted to give about

5% average CLR in the trace-files. About 20% of the losses are dropped packets, the other 80% of the losses are over-delayed packets.

In Table V, for the "Raw" case , the delays are drawn directly from the combined values of Pareto and Gaussian distribution; for the "2 Hz LP" through "0.1 Hz LP" cases, the generated delay signals are filtered with low pass filters with cut-off frequency of 2 Hz, 1 Hz, 0.5 Hz, 0.4 Hz, 0.3 Hz, 0.2 Hz, and 0.1 Hz, respectively; for the "Sin with 1 Hz LP" case, the delay signal is generated using a sinusoidal signal of 0.3 Hz with some white Gaussian noise, and is filtered with a low pass filter of 1 Hz; for the "Sin with 0.4 Hz LP" case, the delay signal is generated using a sinusoidal signal of 0.3 Hz with some white Gaussian noise, and is filtered with a low pass filter of 0.4 Hz.

The predictive path switching control results, in terms of E-model MOS, of seven VoIP packets transmission methods are represented in Table V. The seven VoIP packets transmission methods used are: no switching and transmitting over Path A (A), no switching and transmitting over Path B (B), performing ideal case path switching control (*ideal*), performing predictive path switching control using CLR signals and simple predictors (SP_{CLR}), performing predictive path switching control using delay signals and simple predictors (SP_{delay}), performing predictive path switching control using CLR signals and AR predictors (AR_{CLR}), and performing predictive path switching control using delay signals and AR predictors (AR_{delay}).

The *AR-SP* column gives the MOS difference between the best AR predictors based predictive path switching control result and the best simple predictors based predictive path switching control result

$$\max(AR_{CLR}, AR_{delay}) - \max(SP_{CLR}, SP_{delay}). \tag{5.9}$$

The improvement that can be achieved by performing the ideal case path switch-

ing control is called the ideal case improvement. The *P.R.SP* column gives the percentage realization of the ideal case improvement by the SP method. It shows how much percent of the possible improvement in MOS in the ideal case can be realized by the best SP method

$$P.R.SP = \frac{\max(SP_{CLR}, SP_{delay}) - \max(A, B)}{Ideal - \max(A, B)} \times 100\%. \tag{5.10}$$

The *P.R.AR* column gives the percentage realization of the ideal case improvement by the AR method. It shows how much percentage of the possible improvement in MOS in the ideal case can be realized by the best AR method

$$P.R.AR = \frac{\max(AR_{CLR}, AR_{delay}) - \max(A, B)}{Ideal - \max(A, B)} \times 100\%. \tag{5.11}$$

From the results, the first thing that can be observed is that when the delay signal is generated with a sinusoidal signal plus some white Gaussian noise the AR delay predictor is working significantly better than the simple delay predictor for predictive path switching control. The improvement is 0.64 in MOS in this case. When the delay is drawn from a random distribution, the improvement is not always obvious. The best improvement is 0.22 in MOS in this case. It suggests that when the signal has more deterministic dynamics the predictors developed from dynamic system models will work better than the simple predictors.

The second observation is that in the filtered random delay signal cases, the AR predictors are not much better than simple predictors if the delay signal frequency content is very high (above 1 Hz in this case) or very low (below 0.3 Hz in this case). But in the middle frequency range (around 0.5 Hz in this case) the AR predictors are much better than simple predictors. It suggests that the dynamic system model based predictors are significantly better than simple predictors for predictive path switching control only when some certain range of frequency dominates the signal.

Table V. Emulation study on traffic delay signal frequency content for paths with average CLR of 5%.

Case	Method							AR-SP	P.R.SP	P.R.AR
	A[1]	B[2]	Ideal[3]	SP_{CLR}[4]	SP_{delay}[5]	AR_{CLR}[6]	AR_{delay}[7]		(%)	(%)
Raw	2.63	2.64	3.46	2.67	2.62	2.71	2.67	0.04	3.66	8.54
2 Hz LP	2.54	2.57	3.44	2.56	2.52	2.57	2.61	0.05	-1.15	4.60
1 Hz LP	2.63	2.54	3.55	2.72	2.58	2.80	2.81	0.09	9.78	19.57
0.5 Hz LP	2.55	2.72	3.54	2.80	2.89	2.83	3.11	0.22	20.73	47.56
0.4 Hz LP	2.65	2.62	3.54	2.89	2.89	2.79	3.10	0.21	26.97	50.56
0.3 Hz LP	2.58	2.48	3.50	2.73	3.13	2.75	3.22	0.09	59.78	69.57
0.2 Hz LP	2.61	2.59	3.49	3.03	3.25	3.07	3.27	0.02	72.73	75.00
0.1 Hz LP	2.80	2.53	3.42	3.01	3.19	3.01	3.20	0.01	62.90	64.52
Sin with 1 Hz LP	2.56	2.66	3.57	2.59	2.46	2.67	3.23	0.64	-7.69	62.64
Sin with 0.4 Hz LP	2.60	2.54	3.59	2.72	2.75	2.74	3.39	0.64	15.15	79.80

[1] Transmitting the VoIP packets over Path A only, no switching.

[2] Transmitting the VoIP packets over Path B only, no switching.

[3] Ideal case path switching control.

[4] Predictive path switching control using CLR signals and simple predictors.

[5] Predictive path switching control using delay signals and simple predictors.

[6] Predictive path switching control using CLR signals and AR predictors.

[7] Predictive path switching control using delay signals and AR predictors.

From the percentage realization of the ideal case improvements by SP and AR predictors, it can be seen that as the delay signal frequency is lowered from 2 Hz to 0.1 Hz, the percentage realization of the ideal case improvements has increased from below 10% to around 70%. It shows that when the signal frequency is low, the predictive methods can achieve more of the possible path switching control improvement.

b. The case of 10% comprehensive loss rate

In this set of emulations, the variances of the Pareto distribution and Gaussian distribution, which are combined to generate the delay signals, are adjusted to give about 10% average CLR in the trace-files. About 10% of the losses are dropped packets, the other 90% losses are over-delayed packets.

In Table VI ten emulation cases which are similar to those used in the 5% average CLR emulation study are investigated. The predictive path switching control results, in terms of E-model MOS, of the same seven VoIP packets transmission methods, no switching and transmitting over Path A (A), no switching and transmitting over Path B (B), performing ideal case path switching control (*ideal*), performing predictive path switching control using CLR signals and simple predictors (SP_{CLR}), performing predictive path switching control using delay signals and simple predictors (SP_{delay}), performing predictive path switching control using CLR signals and AR predictors (AR_{CLR}), and performing predictive path switching control using delay signals and AR predictors (AR_{delay}), are represented in Table VI. The MOS difference between the best AR result and the best SP result (*AR-SP*), the percentage realization of ideal case improvement by the best SP method (*P.R.SP*), and that by the best AR method (*P.R.AR*), are also given in the table.

The same trend as in the 5% average CLR emulation set can be observed. When the delay signal is generated by sinusoidal signal plus some white Gaussian noise the

Table VI. Emulation study on traffic delay signal frequency content for paths with average CLR of 10%.

Case	Method							AR-SP	P.R.SP	P.R.AR
	A	B	Ideal	SP_{CLR}	SP_{delay}	AR_{CLR}	AR_{delay}		(%)	(%)
Raw	1.98	2.02	3.05	2.01	1.97	2.13	2.08	0.12	-0.97	10.68
2 Hz LP	1.83	1.74	3.10	1.80	1.75	1.82	1.84	0.04	-2.36	0.79
1 Hz LP	2.00	2.00	3.38	1.98	1.84	2.15	2.14	0.17	-1.45	10.87
0.5 Hz LP	1.70	1.97	3.30	1.95	2.08	1.99	2.31	0.23	8.27	25.56
0.4 Hz LP	1.98	2.01	3.32	2.15	2.34	2.28	2.68	0.34	25.19	51.15
0.3 Hz LP	1.94	2.02	3.35	2.39	2.76	2.36	2.83	0.07	55.63	60.90
0.2 Hz LP	1.74	1.81	3.16	2.29	2.68	2.35	2.80	0.12	64.44	73.33
0.1 Hz LP	1.58	1.83	2.94	2.49	2.72	2.49	2.72	0.00	80.18	80.18
Sin with										
1 Hz LP	1.96	1.99	3.41	1.88	1.87	2.21	2.53	0.65	-7.74	38.03
0.4 Hz LP	1.98	1.93	3.40	2.06	2.16	2.24	2.84	0.68	12.68	60.56

AR delay predictor is significantly better than the SP delay predictor for predictive path switching control. The improvement is more than 0.60 in MOS in these cases. When the delay is drawn from a random distribution, the improvement is not always obvious. The best improvement is 0.34 in MOS in these case.

In the filtered random delay signal cases, the AR predictors are not much better than simple predictors if the signal frequency of delay is very high (above 2 Hz in this case) or very low (below 0.3 Hz in this case). But in the middle frequency range (around 0.5 Hz in this case) the AR predictors are much better than simple predictors.

The percentage realization of the ideal case improvements has increased from less than 10% to about 80%. as the signal frequency is lowered from 2 Hz to 0.1 Hz. One thing to observe is that SP based results are not always better than the best of the two no switching methods. While AR based results are always no worse than the best of the two no switching methods.

The results of the traffic delay signal frequency content study for the 5% average CLR emulations and 10% average CLR emulations suggest that when the signal frequencies are high (over 1 Hz), predictions from both AR predictors SP predictors are bad. The improvement from predictive path switching control is not much. The AR predictors are performing a little better than the SP predictors. When the signal frequencies are in the middle range (about 0.5 Hz), the AR predictors are much better than SP predictors, and the predictive path switching control results given by the AR predictors are much better. When the signal frequencies are very low (below 0.3 Hz), the AR predictors and the SP predictors are performing equally good. The path switching control results are also equally good.

2. Impact of path comprehensive loss rate

In this set of emulations, the impact of path average CLR on predictive path switching control is being studied. The trace-files consist of generated delay signals filtered with a low pass filter of 0.4 Hz cut-off frequency and 1% random losses. In each emulation, both Path A and Path B are assumed to have about the same CLR. Four VoIP packets transmission methods, no switching and transmitting over Path A method (Path A), no switching and transmitting over Path B method (Path B), ideal case path switching control (Ideal switching), and predictive path switching control using delay signal and AR predictor (AR_{delay} predictor), are performed and their results are presented. The CLR covered in this set of emulations are ranging from 1.5% to 30%.

Fig. 14 shows the resulting E-model MOS calculated every 30 seconds of the no switching methods and the AR_{delay} method for the 1.5% average CLR case. It shows that with such a low loss rate, even without switching the voice quality is always above 2.6 MOS. Most of the time there is no need for path switching. It also

shows that the predictive path switching control results are not really better than the best no switching results, but the predictive path switching control results are more consistent, while the no switching results of one path can vary a lot.

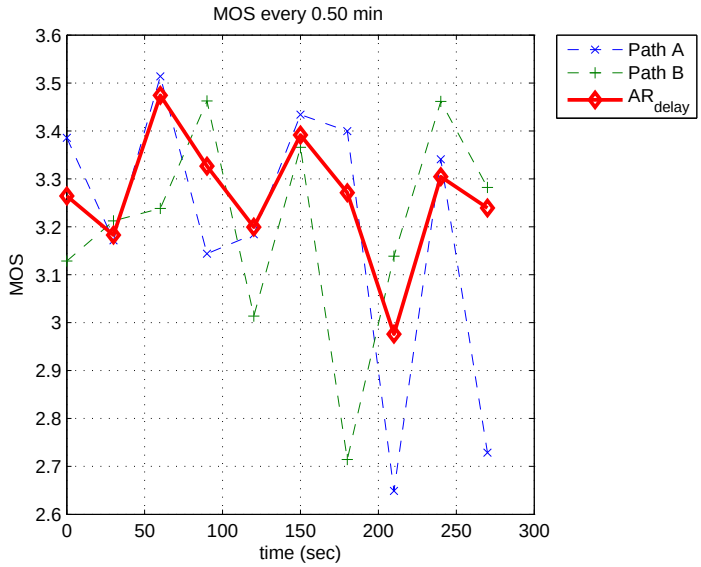

Fig. 14. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 1.5% average CLR emulation case.

Fig. 15 shows the resulting E-model MOS calculated every 30 seconds of the no switching methods and the AR_{delay} method for the 2% average CLR case. This time as the loss rate has increased, it can be seen that the voice quality in the two paths are not as good as in the previous case, but most of the results are still above 2.6 MOS. In this case, the predictive path switching control result is better than the best of no switching results most of the time. It would make more sense to start using predictive path switching control from this point on.

Fig. 16 shows the resulting E-model MOS calculated every 30 seconds of the no switching methods and the AR_{delay} method for the 5% average CLR case. It shows

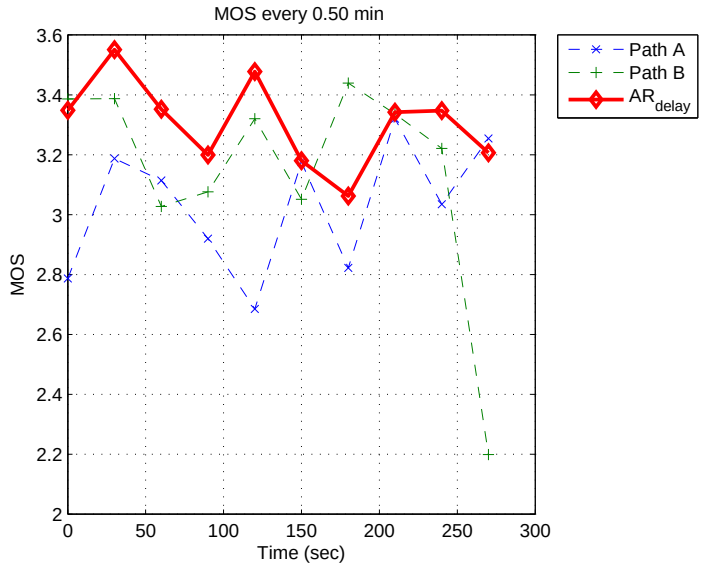

Fig. 15. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 2% average CLR emulation case.

that for the no switching methods, half of the time their results are unacceptable. The predictive path switching control results are better than the best of no switching results most of the times. And the predictive path switching control results are mostly in the acceptable range. But the result MOS is worse than it is in the 2% CLR case. Using predictive path switching is a good choice in this case.

Fig. 17 shows the resulting E-model MOS calculated every 30 seconds of the no switching methods and the AR_{delay} method for the 10% average CLR case. Now, except for one point, all the no switching results are below 3 MOS. And except for that point, the predictive path switching control results are the best of all at each time. Without using the predictive path switching control, the voice quality is unacceptable from this point on.

Fig. 18 shows the resulting E-model MOS calculated every 30 seconds of the no

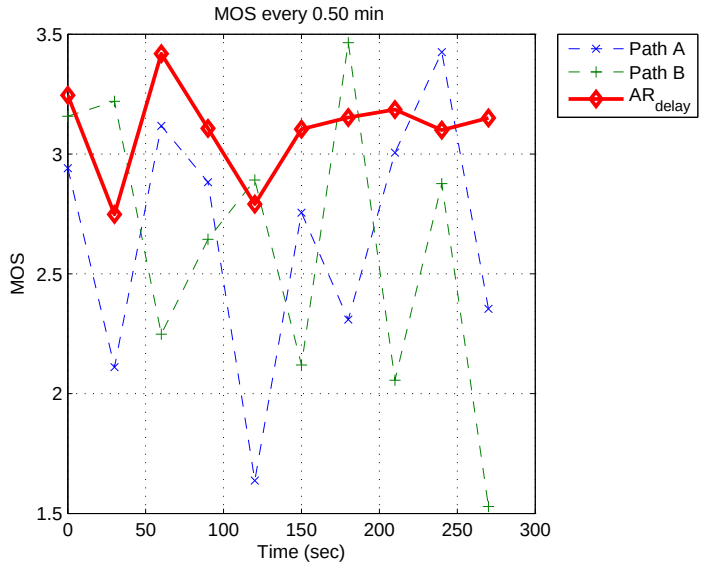

Fig. 16. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 5% average CLR emulation case.

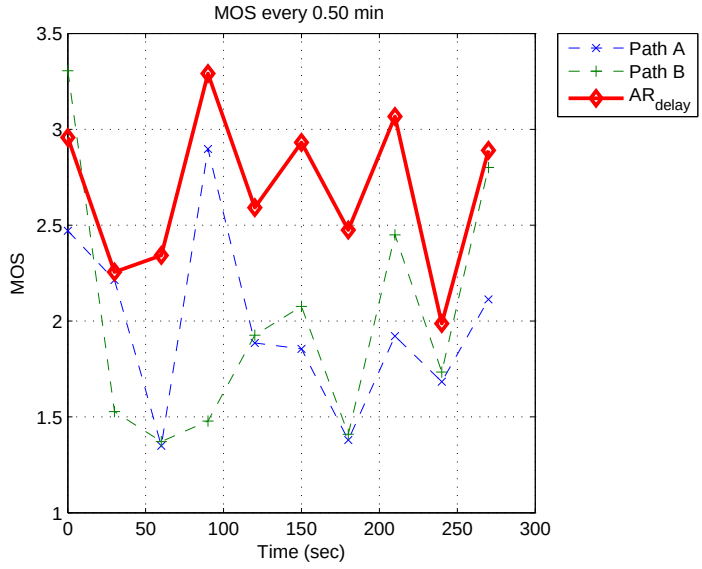

Fig. 17. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 10% average CLR emulation case.

switching methods and the AR_{delay} method for the 15% average CLR case. In this case, the no switching results are all below 2.5 MOS, the voice quality is very bad. The predictive path switching control results are better than the best of no switching results at all time. But even the predictive path switching control results are only a little above 2 MOS. So even if using this control, the voice quality is bad.

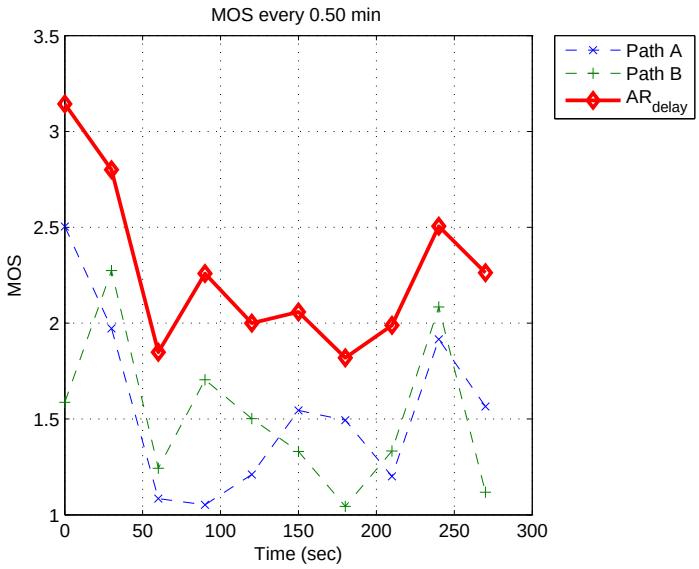

Fig. 18. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 15% average CLR emulation case.

Fig. 19 shows the resulting E-model MOS calculated every 30 seconds of the no switching methods and the AR_{delay} method for the 20% average CLR case. In this case, the no switching results are all below 1.5 MOS, the voice quality is very bad. The predictive path switching control results are better than the best of no switching results at all time. But even the predictive path switching control results are also mostly below 2 MOS. So even if using this control, the voice quality is very bad.

Fig. 20 shows the resulting E-model MOS calculated every 30 seconds of the no

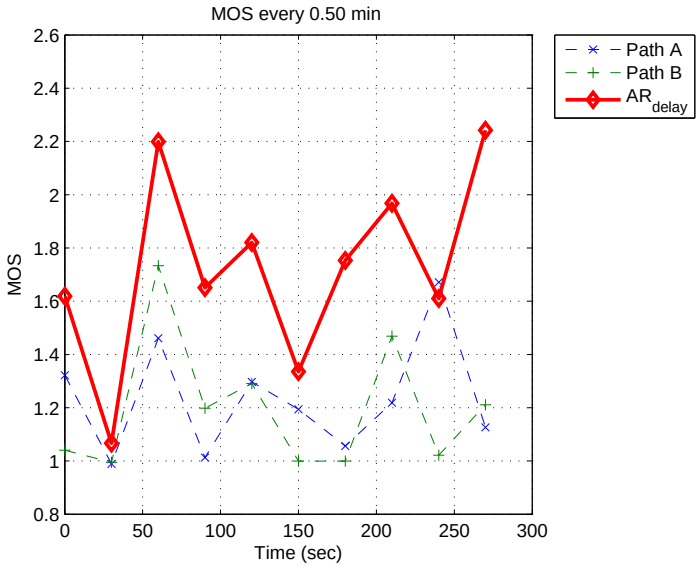

Fig. 19. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 20% average CLR emulation case.

switching methods and the AR_{delay} method for the 30% average CLR case. In this case, the no switching results are all close to 1 MOS, the VoIP application makes no sense in this case. The predictive path switching control results are better than the best of no switching results at all time. But even the predictive path switching control results are all below 2 MOS. So even if using this control, the resulting voice is useless.

Table VII shows the resulting E-model MOS of, the no switching and transmitting over Path A method, the no switching and transmitting over Path B method, the ideal case path switching control , and the predictive path switching control using delay signal and AR predictor (AR_{delay}), under each average path CLR. The absolute amount of improvement in terms of MOS by AR_{delay} method over the best no switching method is also given in the table in the row "Improvement by AR_{delay}".

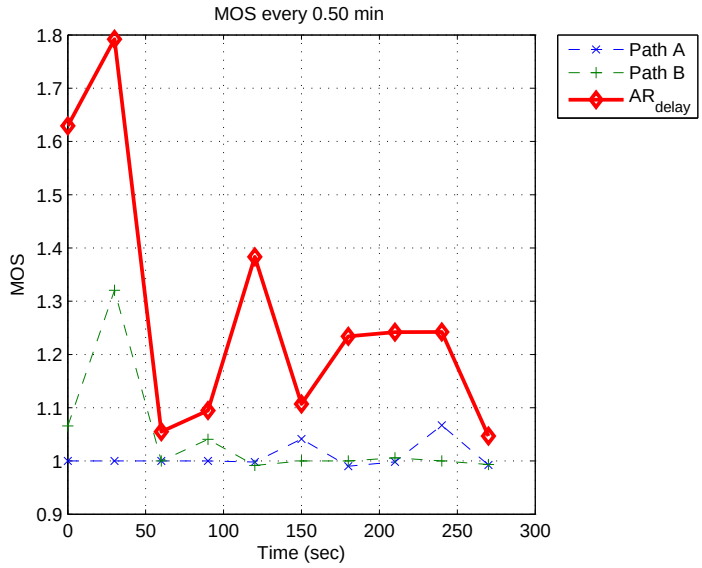

Fig. 20. Resulting E-model MOS every 30 seconds of no switching methods and the AR_{delay} method for a 30% average CLR emulation case.

It can be observed that when the average CLR of both paths are below 2%, even without path switching the MOS of both paths are above 3, thus the voice quality will be in the fair range. There is no need to perform any VoIP QoS control in this case. When the average CLR of both paths are above than 15%, even using the ideal case path switching control the resulting MOS is below 3, thus the voice quality is poor. In this case, predictive path switching control cannot improve the voice quality to an acceptable level. So the path switching method is more meaningful when the average CLRs of the two paths are above 2% and below 15%. It can also be observed that the improvement of the predictive path switching method over the no switching methods is insignificant when the CLR of the path is very low or very high. There is more significant improvement when the average CLR is in the range above 5% and below 20%. This result suggests that the predictive path switching control method

Table VII. Emulation study on average path CLR

Method	E-model MOS results on path pairs with average CLR of						
	1.5%	2%	5%	10%	15%	20%	30%
Path A - No Switching	3.20	3.03	2.65	1.98	1.55	1.23	1.01
Path B - No Switching	3.20	3.14	2.62	2.01	1.52	1.20	1.04
Ideal Switching	3.57	3.57	3.54	3.32	3.14	2.46	1.85
AR_{delay}	3.26	3.31	3.10	2.68	2.27	1.73	1.28
Improvement by AR_{delay}	0.06	0.17	0.45	0.67	0.72	0.50	0.24

can give the best improvement when the average CLRs of the two paths are in the 5% to 15% range.

E. Chapter Summary and Emulation Study Conclusions

In this chapter, the method of using predictive path switching control to improve the VoIP QoS has been introduced. The problem statement has been given. Dynamic system model based predictors have been proposed. The general assumptions behind this method has also been discussed.

The emulation study on the impact of traffic delay signal frequency content shows that the more deterministic the network condition is, the better the dynamic system model based predictors are performing over the simple predictors for the predictive path switching control. Because if the network condition has more deterministic characteristic, then the dynamic system models can captured the deterministic dynamics of the system and can give better predictions compared to the simple predictor, which lead to better path switching control performance. If the network condition is less deterministic, then the performance will depend on the frequency content of the traffic delay signals. If the delay signal frequency content is very high (above 1 Hz), then

both AR predictors and the simple predictors cannot capture the dynamics in the system, and both give bad predictions which lead to bad predictive path switching control results. In this case the AR predictors give no significant improvement over the simple predictors, as their performances are equally bad. If the delay signal frequency content is very low (below 0.3 Hz), then both AR predictors and the simple predictors can capture the dynamics in the system, and both give good predictions which lead to good predictive path switching control results. In this case the AR predictors give no significant improvement over the simple predictors, as their performances are equally good. If the delay signal frequency content is in the middle range (about 0.5 Hz), then the AR predictors can capture some of the system dynamics which is not captured by the simple predictors. In this case the AR predictors can give better predictions than the simple predictors, thus the performance of the predictive path switching control based on AR predictors is much better than that of the predictive path switching control based on simple predictors. But for both the AR predictors based predictive path switching control and the simple predictors based predictive path switching control, the lower the frequency content of the delay signals, the better the performance of the controls.

The emulation study on the impact of path loss rate shows that below average CLR of 2% , there is no need of any VoIP QoS control, as the voice quality is good enough even without any control, and above average CLR of 15%, even the ideal case path switching control cannot give enough improvement to the voice quality to make it acceptable. Thus using path switching control to improve the VoIP QoS is only meaningful when the average CLR of both paths are above 2% and below 15%. The emulation study also shows that predictive path switching control have more significant improvement over no switching methods when the average CLR is in the range of 5% to 20%. So the predictive path switching control is likely to give the best

VoIP QoS improvement when the average CLR of the two paths is in the 5% to 15% range.

CHAPTER VI

ACTUAL NETWORK DATA COLLECTION AND PRELIMINARY INVESTIGATION ON PATH SWITCHING CONTROL

This chapter discusses the problems encountered in collecting data from an actual network. Then it gives a preliminarily investigation of the possibility of improving the VoIP QoS through path switching using actual network data.

The rest of this chapter is organized as the following. Section A summarizes the data collection method, the problems in data collection, and their solutions. Section B shows the preliminary investigation results of the path switching control with actual network data. The CLR ranking changes of the paths, the results of ideal case path switching, and the results of the simple predictor (SP) based path switching are investigated. Section C concludes the results in this chapter.

A. Data Collection

In this section, first the method used to collect actual network data for this study is discussed. Some of the problems coming up in data collection are addressed. The solutions to these problems are also given. The comprehensive loss rate (CLR) is used as a preliminary criterion for path quality estimation.

1. PlanetLab

As the effectiveness of path switching highly depends on the temporal difference in the trace-files of different paths [50], artificially generated profiles and actual network profiles can give very different results. So trace-file data are collected from actual network paths from the PlanetLab.

PlanetLab is an open platform for developing, deploying, and accessing planetary-

scale services. It currently consists of 665 machines, hosted by 315 sites, spanning over 25 countries [109]. It is capable of serving as a test-bed for overlay networks. The applications running on PlanetLab nodes work under "real-world" conditions. So the trace-files of different paths used in this study are collected from real host pairs on the PlanetLab test-bed.

2. Probing

In this study, without loosing the generality, it is assumed that the VoIP applications use the Speex codec for encoding their VoIP packet. The minimum frame size of the Speex codec is 20 ms, and the lowest bit rate of the Speex codec is about 5 kilobits per second (kbps). In most measurements, the path between a pair of nodes is probed by UDP packets of 38 bytes with 100 millisecond (ms) inter-departure time, which is mimicking the behavior of a VoIP application using the lowest 5 kbps bit rate. In some measurements, probing packets with inter-departure time of 20 ms were used to compare with the 100 ms cases. The trace-files were obtained from the records of these probing packets.

If a probing packet is directly sent from the source to the destination and is echoed back from the destination directly to the source, then the send time stamp of this packet is recorded with its sequence number (SN) at the source, the receive time stamp of this packet is recorded with its SN at the destination, when this probing packet is echoed back from the destination to the source, the send time stamp and the receive time stamp of this echoed packet are also recorded with its SN at the destination and the source, respectively. All four records are used to find out the delays and losses in both forward and backward directions on the path of Internet.

If a probing packet is first sent from the source to an overlay node letting the overlay node forward it to the destination, then it is echoed back from the destination

back to the the overlay node letting the overlay node forward it back to the source, then, besides the four time stamps recorded at the source and the destination as in the previous direct sending case, the arrive time stamp and forward time stamps of the probing packet at the overlay node and those time stamps of the echoed packet at the overlay node are also recorded with its SN at the overlay node. Through analysis of these eight time stamps, the networks' condition in the source to the overlay node path and the networks' condition in the overlay node to the destination path can be obtained.

3. Clock skew

One problem that shows up in the path trace-file data collection is the clock differences at the source and destination. In one-way delay measurements, as done in this research, time stamps from two different machines are used. But the two clocks from these two machines are usually not well synchronized. Particularly, the two clocks may have different speeds. The difference in the clock speed is called the *clock skew*. Clock skew will result in a system error in the delay measurement. The top plot of Fig. 21 is an example of the uncorrected delay measurement that will result if the clock skew is not removed. The clock skew is reflected by the trend in the line formed at the bottom of the uncorrected delay measurements.

Paxson [88] studied this problem, and suggested some batch process methods to determine the presence of clock skew. After comparing different clock skew estimation methods, Moon et al. [89] recommended the linear programming method. Zhang et al. [90] suggested a convex hull approach for clock skew estimation. Bletsas [91] suggested an algorithm based on Kalman filtering. The Network Time Protocol (NTP) [86] is designed to solve this problem by synchronizing the clocks of computer systems over networks. Melvin and Murphy [85] discussed how time syn-

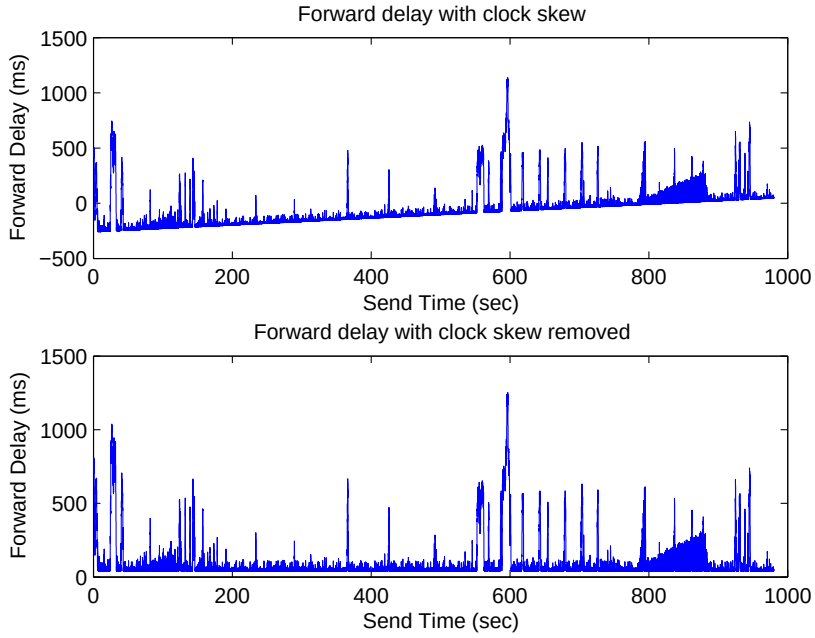

Fig. 21. Effect of time synchronization on delay measurement. Top: Uncorrected forward delay with clock skew. Bottom: Corrected forward delay without clock skew.

chronization can help to improve the VoIP QoS. Johannessen [87] gave a good review on the time synchronization problem. In this study Zhang's convex hull approach is used for clock skew estimation. The trace-files are obtained after removing the clock skew from the time stamps.

Assume the time measurement from the sender side clock is T_1, and the the time measurement from the receiver side clock is T_2. The sender side clock is taken as the reference clock, which is assumed to be the true time measurement

$$T_1 \equiv t. \tag{6.1}$$

Then comparing to the sender side clock, the time measurement from receiver side clock for the same time point will have a skew $\alpha - 1$ and an offset β.

$$T_2 = \alpha t + \beta. \tag{6.2}$$

Assume that the sender sends a probing packet at time t_1 with the sender side clock time stamp t_{s1}, it arrives at the receiver at t_2 with the receiver side clock time stamp t_{r2}, the receiver echos this probing packet at time t_3 with the receiver side clock time stamp t_{r3}, and the echoed packet arrives at the sender at time t_4 with the sender side clock time stamp t_{s4}. Assume that the true forward delay is $d_1(t_1)$, the true backward delay is $d_2(t_3)$, and the response time at the receiver is $\tau(t_2)$.

The relation of these time stamps, as shown in Fig. 22, are governed by the following equations:

$$t_{s1} = t_1, \tag{6.3}$$

$$t_{r2} = \beta + \alpha t_2 = \beta + \alpha \left(t_1 + d_1 \left(t_1\right)\right), \tag{6.4}$$

$$t_{r3} = \beta + \alpha t_3 = \beta + \alpha \left(t_1 + d_1 \left(t_1\right) + \tau \left(t_2\right)\right), \tag{6.5}$$

$$t_{s4} = t_4 = t_1 + d_1 \left(t_1\right) + \tau \left(t_2\right) + d_2 \left(t_3\right). \tag{6.6}$$

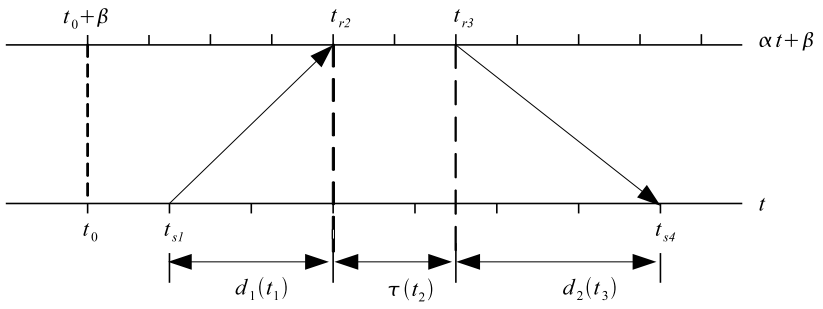

Fig. 22. Time stamp relations in delay measurements.

The uncorrected forward delay is

$$\begin{aligned} t_{r2} - t_{s1} &= \beta + \alpha\left(t_1 + d_1\left(t_1\right)\right) - t_1 \\ &= \left(\alpha - 1\right) t_{s1} + \alpha d_1\left(t_1\right) + \beta \\ &= m_1 t_{s1} + c_1 \end{aligned} \tag{6.7}$$

where, m_1 and c_1 are the slope and offset of the line formed by the bottom of the uncorrected forward delay measurements $t_{r2} - t_{s1}$ with respect to send time t_{s1}, respectively.

The uncorrected backward delay is

$$\begin{aligned} t_{s4} - t_{r3} &= t_3 + d_2\left(t_3\right) - \beta - \alpha t_3 \\ &= \left(1 - \alpha\right) t_3 + d_2\left(t_3\right) - \beta \\ &= \left(1 - \alpha\right)\left(t_4 - d_2\left(t_3\right)\right) + d_2\left(t_3\right) - \beta \\ &= \left(1 - \alpha\right) t_{s4} + \alpha d_2\left(t_3\right) - \beta \\ &= m_2 t_{s4} + c_2. \end{aligned} \tag{6.8}$$

where, m_2 and c_2 are the slope and offset of the line formed by the bottom of the uncorrected backward delay measurements $t_{s4} - t_{r3}$ with respect to arrive time t_{s4},

respectively.

Assume a collection of measurement data has been obtained

$$\Omega := \left\{ v_i = (\tilde{t_i}, \tilde{d_i}) : i = 1, \ldots, N \right\},$$

where $\tilde{t_i}$ is the send time , $\tilde{d_i}$ is the uncorrected delay, and v_i is the send time and uncorrected delay pair. The problem of estimating the slope m and offset c from line formed by the bottom of the delay ($\tilde{d_i}$) vs send time ($\tilde{t_i}$) plot can be formulated as a Linear Programming (LP) problem [89]

$$\begin{aligned} &\min_{a,b \in \mathbb{R}} \left\{ \sum_{i=1}^{N} \left(\tilde{d_i} - m\tilde{t_i} - c \right) \right\} \\ &\tilde{d_i} - m\tilde{t_i} - c \geq 0, \qquad 1 \leq i \leq N \end{aligned} \quad , \tag{6.9}$$

where $\mathbb{R}$ is the set of real numbers. The constraint indicates that all points in Ω are above the straight line $L = \{(x, y) | y = mx + c\}$, and the minimization ensures that the obtained straight line L has the minimum distance to all the send time and uncorrected delay points (pairs). Thus m and c are the slope and offset of the line formed by the bottom of the delay vs send time plot. Define the convex hull of N points as a polytope enclosed by piecewise linear functions, then the constraint in (6.9) is equivalent to say that the the convex hull of Ω,

$$\mathrm{co}(\Omega) := \left\{ x | x = \sum_i \lambda_i v_i, \lambda_i \geq 0, \sum_i \lambda_i = 1, v_i \in \Omega \right\} \tag{6.10}$$

is above L. The "closest" line L to Ω will touch Ω at some point [90]. The solution to equation (6.9) is the section of the lower boundary of co(Ω) that covers the point $\sum_i \frac{t_i}{N}$ [90].

Zhang's algorithm to calculate the convex hull of Ω is given below.

Algorithm Convex_Hull_L:

1. Initialize:

   Prepare an empty stack which can store the send time and uncorrected delay pairs ($\tilde{t}$,$\tilde{d}$), which will be points on the send time vs uncorrected delay plot.

   Push the first send time and uncorrected delay pair (point), $v_1 = (\tilde{t_1}, \tilde{d_1})$, into the stack;

   Push the second send time and uncorrected delay pair (point), $v_2 = (\tilde{t_2}, \tilde{d_2})$, into the stack;

2. For $i = 3$ to N

   If ($v_i = (\tilde{t_i}, \tilde{d_i})$ is above $line(top, next\_to\_top)$)

   push the send time and uncorrected delay pair (point), v_i, into the stack;

   Else

   While (v_i is below $line(top, next\_to\_top)$ and $stack\_size > 1$)

   Pop the top send time and uncorrected delay pair (point) from the stack;

   End while

   Push the send time and uncorrected delay pair (point), v_i, into the stack;

   End if

   End for

3. End

where $line(v, w)$ is used to denote the straight line connecting the two points v and w, $stack\_size$ is the stack size.

Using Zhang's convex hull algorithm, the slope m_1 and offset c_1 of the line formed by the bottom of the uncorrected forward delay measurements $t_{r2} - t_{s1}$ with respect to send time t_{s1}, and the slope m_2 and offset c_2 of the line formed by the bottom of the uncorrected backward delay measurements $t_{s4} - t_{r3}$ with respect to arrive time t_{s4}, can be obtained.

Once m_1, c_1, m_2, and c_2 are obtained using Zhang's convex hull algorithm, from the data points on those lines, i.e. the data points with the minimum Round Trip Time (RTT), the following relations can be obtained from equation (6.7) and (6.8):

$$m_1 = \alpha - 1 \quad \Rightarrow \quad \alpha = m_1 + 1, \tag{6.11}$$

$$c_1 = \alpha d_1(t_1) + \beta \quad \Rightarrow \quad \beta = c_1 - \alpha d_1(t_1), \tag{6.12}$$

$$m_2 = 1 - \alpha \quad \Rightarrow \quad \alpha = 1 - m_2, \tag{6.13}$$

$$c_2 = \alpha d_2(t_3) - \beta \quad \Rightarrow \quad \beta = -c_2 + \alpha d_2(t_3). \tag{6.14}$$

If the minimum forward delay is also assumed to be equal to the minimum backward delay, i.e. $d_1(t_1) = d_2(t_3)$, then from equation (6.11) and (6.13), the estimate of α is given by

$$\hat{\alpha} = \frac{a_1 - a_2 + 2}{2}, \tag{6.15}$$

where, $\hat{\alpha}$ is the estimate of α; and from equation (6.12) and (6.14), the estimate of β is given by

$$\hat{\beta} = \frac{b_1 - b_2}{2}, \tag{6.16}$$

where $\hat{\beta}$ is the estimate of β.

From equation (6.3) and (6.4), the corrected forward delay is given by:

$$t_{s1} = t_1 \quad \Rightarrow \quad \hat{t}_1 = t_{s1} \tag{6.17}$$

$$t_{r2} = \beta + \alpha(t_1 + d_1(t_1)) \quad \Rightarrow \quad t_{r2} - \beta = \alpha(t_1 + d_1(t_1))$$

$$
\begin{aligned}
&\Rightarrow \quad t_1 + d_1(t_1) = \frac{t_{r2} - \beta}{\alpha} \\
&\Rightarrow \quad d_1(t_1) = \frac{t_{r2} - \beta}{\alpha} - t_1 \\
&\Rightarrow \quad d_1(t_1) = \frac{t_{r2} - \beta}{\alpha} - t_{s1} \\
&\Rightarrow \quad \hat{d}_1(\hat{t}_1) = \frac{t_{r2} - \hat{\beta}}{\hat{\alpha}} - t_{s1} \qquad (6.18)
\end{aligned}
$$

where $(\hat{t}_1, \hat{d}_1(\hat{t}_1))$ is the corrected send time and forward delay pair.

From equation (6.5) and (6.6), the corrected backward delay is given by:

$$
\begin{aligned}
t_{r3} = \beta + \alpha t_3 \quad &\Rightarrow \quad t_{r3} - \beta = \alpha t_3 \\
&\Rightarrow \quad t_3 = \frac{t_{r3} - \beta}{\alpha} \\
&\Rightarrow \quad \hat{t}_3 = \frac{t_{r3} - \hat{\beta}}{\hat{\alpha}} \qquad (6.19) \\
t_{s4} = t_1 + d_1(t_1) + \tau(t_2) + d_2(t_3) \quad &\Rightarrow \quad t_{s4} = t_3 + d_2(t_3) \\
&\Rightarrow \quad d_2(t_3) = t_{s4} - t_3 \\
&\Rightarrow \quad d_2(t_3) = t_{s4} - \frac{t_{r3} - \beta}{\alpha} \\
&\Rightarrow \quad \hat{d}_2(t_3) = t_{s4} - \frac{t_{r3} - \hat{\beta}}{\hat{\alpha}} \qquad (6.20)
\end{aligned}
$$

where $(\hat{t}_3, \hat{d}_2(\hat{t}_3))$ is the corrected send time and forward delay pair. The bottom plot of Fig. 21 shows the corrected delay measurement when the clock skew is removed from the top plot.

4. Preliminary path quality criterion

The voice quality of a VoIP application is affected by both the packet loss rate and the packet delay it has experienced. The comprehensive loss rate (CLR) gives the probability of a packet either getting lost (or is dropped) in the network, or exceeding the one-way delay limit, being dropped at the destination, and being considered lost.

As a rule of thumb, for VoIP applications, when the VoIP packet loss rate is below $1\% \sim 2\%$, the voice quality will still be very good [152], there is no need for VoIP QoS control. When the packet loss rate is over 5%, the VoIP quality will be bad [153]. And when the packet loss rate exceeds 20%, the VoIP quality is degraded beyond use [99]. So CLR is used as the preliminary criterion for path quality estimation in this study.

5. Choice of paths

a. Overlay paths

A limitation of PlanetLab at the time the data collection was conducted, was that each PlanetLab node had only one connection to the Internet. So at the beginning, the overlay structure [5] was tried to form a path diversity.

Between a pair of nodes, one path is formed by directly sending the packets from the source to the destination and then bouncing them back from the destination directly to the source. An alternative path is formed by first sending the packets from the source to an overlay node and letting the overlay node to forward the packets to the destination, then letting the destination responds the packets back to the overlay node and letting the overlay node forward the responses back to the source. The overlay structure is show in Fig. 23.

Probing packets were send at 100 ms intervals at the same time for both the direct and overlay paths. The sequence numbers (SNs) and time stamps of the packets were recorded at the source, the destination, and the overlay nodes. Then the delays and loss rates of each link were calculated.

It turned out that for the PlanetLab nodes, the trace-file of the alternative path contains all the delay spikes of the direct path, plus some spikes of its own. The top

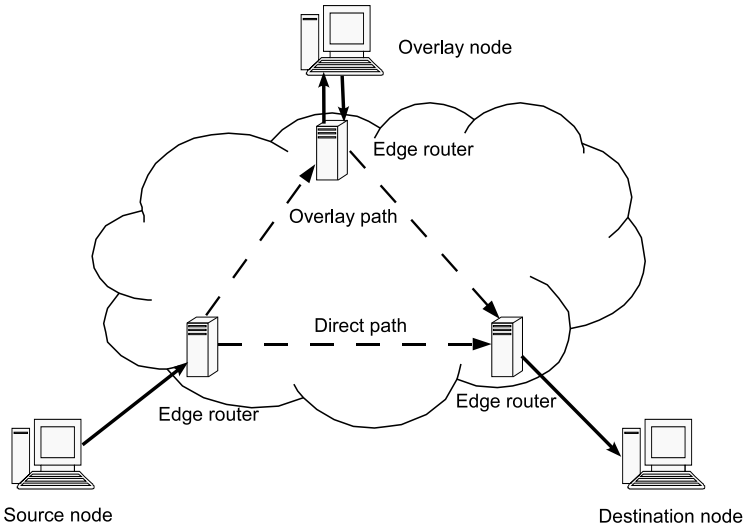

Fig. 23. An overlay structure.

and middle plots of Fig. 24 shows an example of the two trace-files. The bottom plot shows the delay difference of the two trace-files. As the difference is always positive, it means the alternative path had always had more delays than the direct path. This might be an indication that most of the delays in the trace-file are developing at the edge routers instead of the backbone routers. As there is no link to an alternative edge host for PlanetLab hosts, two paths with suitable trace-files for path switching between a pair of nodes cannot be found through an overlay structure in PlanetLab.

b. Two path

As there is a problem with getting a suitable alternative path with the overlay structure of the PlanetLab, two different paths between two different pairs of nodes were attempted instead. Assume that there is an imaginary source station which takes the two source hosts as its two edge routers to the Internet from two different Internet Service Providers (ISPs). Likewise assume that there is an imaginary destination

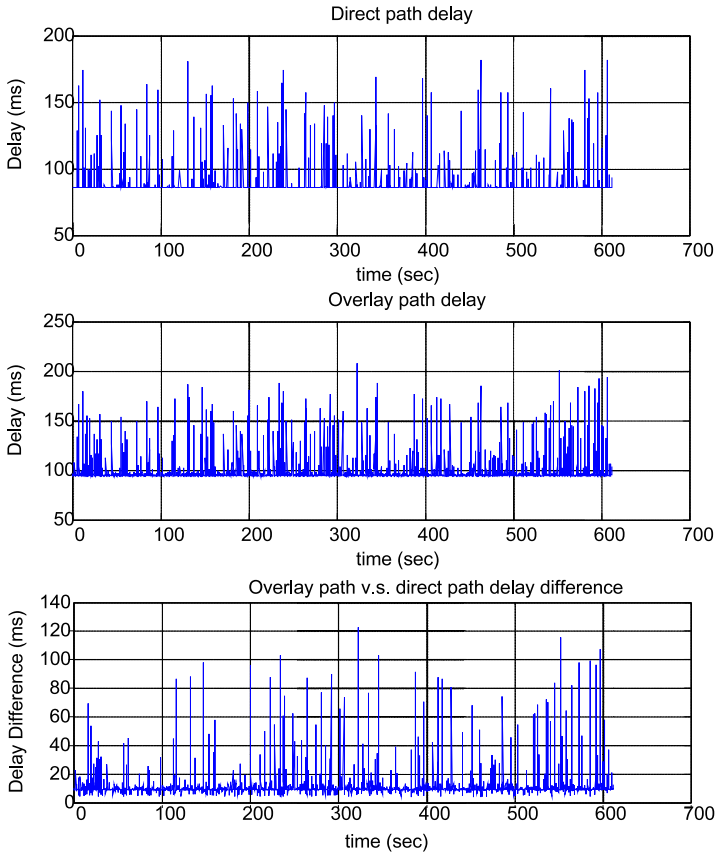

Fig. 24. Trace of a direct path and an overlay path. The direct path is between *pli1-pa-3.hpl.hp.com* node (source) and *planetlab1.informatik.uni-kl.de* node (destination). The overlay node is *planetlab1.csail.mit.edu*. Top: trace of the direct path. Middle: trace of the overlay (alternative) path. Bottom: the delay difference between the alternative path and the direct path.

station which takes the two destination hosts as its two edge routers to the Internet from two different ISPs. Then when probing from this imaginary source station to the imaginary destination station for the two paths, the same trace-files will be obtained, as probing at the two separate paths simultaneously from the PlanetLab source hosts to the PlanetLab destination hosts. Fig. 25 shows this scenario. Fig. 26 shows an example of the trace-files of the two paths and their delay difference. The top plot shows the trace of the path from the *thu1.6planetlab.edu.cn* node (source) to the *pli1-pa-3.hpl.hp.com* node. The "+" signs in the plot designate the lost packets, the solid lines designate the delays of of the probing packets. The middle plot shows the trace of the path from the *fudan1.6planetlab.edu.cn* node (source) to the *planet1.seattle.intel-research.net* node (destination) path. And the bottom plot shows the delay differences between the two paths. This time it can be clearly seen that the differences in delay between the two paths have both positive and negative values, which indicates no one path is consistently better than the other. Thus path switching is feasible.

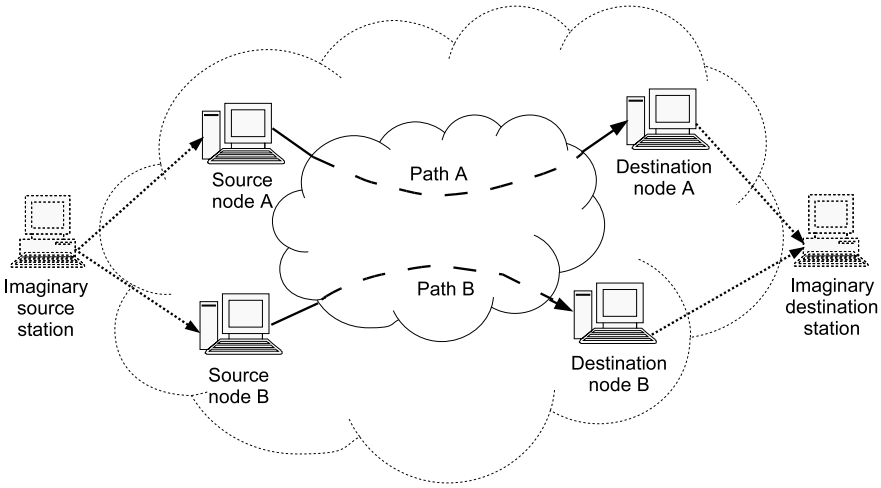

Fig. 25. A scenario of the two paths scheme.

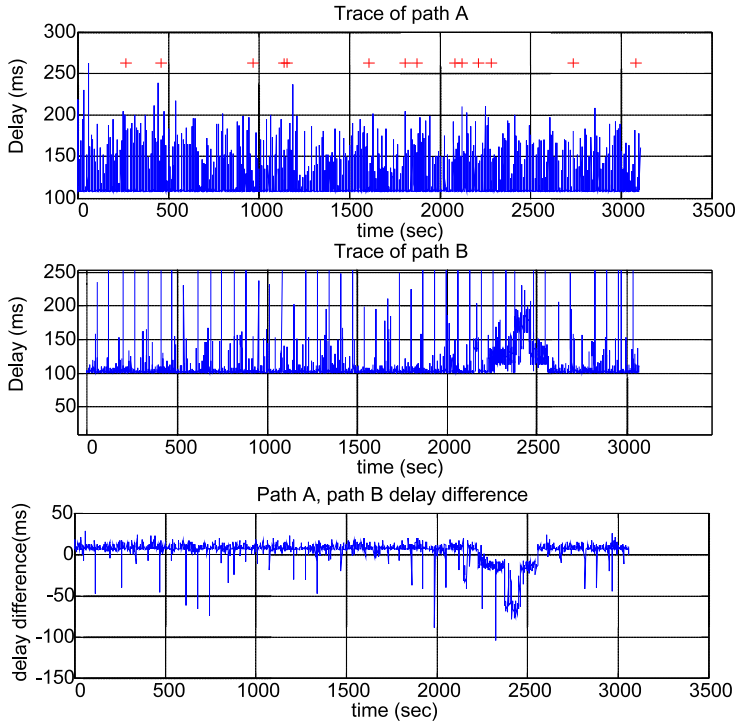

Fig. 26. Trace of two paths. Top: trace of the *thu1.6planetlab.edu.cn* node (source) to *pli1-pa-3.hpl.hp.com* node (destination) path. The"+" signs designate lost packets. The solid lines designate the delays of the probing packets. Middle: trace of the *fudan1.6planetlab.edu.cn* node (source) to *planet1.seattle.intel-research.net* node (destination) path. Bottom: the delay difference between the two paths.

When the CLRs for both paths are very low, such as $1 \sim 2\%$, a VoIP application running through any of the paths will have almost toll quality. There will be no need for path switching. On the other hand if both paths have very high CLRs, then whichever path being switched into, the voice quality will still be very bad. So path switching can make a significant difference in VoIP quality, if two paths with CLRs between $5\% \sim 20\%$ are found.

The CLRs calculated over a one hour interval for some of the paths measured during May, June, and July, 2005, are listed in Table VIII. Of all the PlanetLab host pairs that have been tried within the U.S., only the path between *planetlab1.nbgisp.com* node to *planetlab1.gti-dsl.nodes.planet-lab.org* node has shown a comprehensive loss rate over 5%. These two nodes are both Digital Subscriber Line (DSL) nodes. The other node pairs showed comprehensive loss rates no more than $1 \sim 2\%$. Thus there is no need for control to improve the VoIP QoS on these paths. For overseas nodes, the paths from U.S. nodes to European nodes also showed comprehensive loss rates less than 1%. On the other hand some paths to Indian nodes showed comprehensive loss rates over 50% (mainly due to the high delay), which are probably beyond the possibility of improvement by a path switching control. Thus there is some difficulty in finding two suitable paths to attempt dynamic path switching.

c. One path

As only the path from *planetlab1.nbgisp.com* node to *planetlab1.gti-dsl.nodes.planet-lab.org* node has been found to have a comprehensive loss rate over 5%, data collected from this path on different days are taken as data from different paths for this study.

Three sets of data have been chosen. Each data set has 9 trace-files from the same day. Each trace-file has recorded one hour of probing results. Each data set is taken as a separate path. The first data set, taken as Path A, has an average CLR

Table VIII. CLRs of some actual network paths.

	source	destination	CLR (%)	
US to Asia	hplhp[1]	cuhk[2]	0.21	
	nbgisp[3]	ntutw[4]	0.08	
	hplhp	thu[5]	1.11	
	seaintel[6]	fudan[7]	0.70	
Asia to US	cuhk	hplhp	1.32	
	ntutw	nbgisp	1.23	
	thu	hplhp	0.13	
	fudan	seaintel	0.19	
US	mit[8]	stanford[9]	0.22	
	pbs[10]	hplhp	1.62	
	crldec[11]	seaintel	0.26	
	nbgisp	gtidsl[12]	5.37	
	seaintel	crldec	0.61	
Europe	hplhp	informatik[13]	0.54	
	informatik	hplhp	0.06	
India	nussg[14]	iiitbin[15]	99.86	(42.48[16])
	iiitbin	nussg	100	(25.31[16])
	gtidsl	iiitbin	100	(42.28[16])

[1] pli1-pa-3.hpl.hp.com (US node)

[2] planetlab1.ie.cuhk.edu.hk (Asian node)

[3] planetlab1.nbgisp.com (US node)

[4] planetlab1.im.ntu.edu.tw (Asian node)

[5] thu1.6planetlab.edu.cn (Asian node)

[6] planet1.seattle.intel-research.net (US node)

[7] fudan1.6planetlab.edu.cn (Asian node)

[8] planetlab1.csail.mit.edu (US node)

[9] planetlab-1.stanford.edu (US node)

[10] planetlab1.pbs.org (US node)

[11] pli1-crl-1.crl.dec.com (US node)

[12] planetlab1.gti-dsl.nodes.planet-lab.org (US node)

[13] planetlab1.informatik.uni-kl.de (European node)

[14] soccf-planet-001.comp.nus.edu.sg (Asian node)

[15] planetlab1.iiitb.ac.in (Indian node)

[16] The CLR in parentheses is calculated for the case when the one-way delay limit is relaxed to 300 ms.

of 11%. The second data set, taken as Path B, has an average CLR of 20%. The third data set, taken as Path C, has an average CLR of 9%. Thus a study of the path pair AB represents a case where the two paths have a large qualitative difference. A study of the path pair AC represents a case where the two paths have about the same quality. Sample trace-files of Path A, Path B, and Path C are presented in Fig. 27. In the figure, the "+" signs designate losses, the solid lines designate the delays, and the dash-dotted lines designate the one-way delay threshold.

B. Preliminary Path Switching Results

1. Number of ranking changes

Tao et al. concluded in their study that "substantial improvements can be realized (by path switching) in lowering end-to-end losses, while the benefits for end-to-end delay are typically marginal except in a few rare instances" [50]. In this study, the CLR, which is reflecting the effect of both end-to-end losses and end-to-end delay, is the preliminary criterion of voice quality. It needs to be checked that if path switching will improve the end-to-end CLR and in turn improve the voice quality.

To explore the possible benefits of path switching, first the end-to-end CLRs of different paths are compared. For each path, its CLR is calculated every 400 ms. For a pair of two paths, their quality is ranked according to their CLRs every 400 ms. It is checked that in every hour how many times has the ranking changed, and that in each hour what's the percentage of time for each one path to be better than the other. High number of CLR ranking changes and evenly distributed percentage of time for each path to be better than the other, indicates a good chance of performance improvement using path switching.

Table IX shows the number of CLR ranking changes every hour (each trace-file

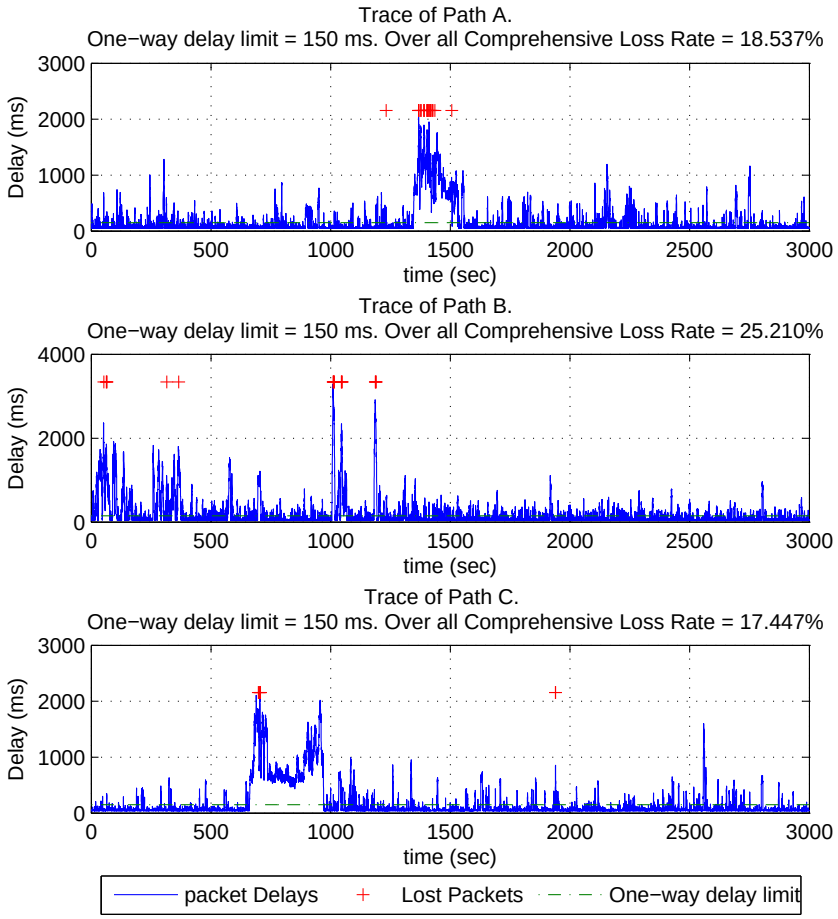

Fig. 27. Sample trace-files of Path A, Path B, and Path C. Top: a sample trace of Path A. Middle: a sample trace of Path B. Bottom: a sample trace of Path C. The"+" signs designate lost packets. The solid lines designate the delays of the probing packets. The dash-dotted lines designate the one-way delay threshold of 150 ms.

has one hour of data) for each path pairs, and the percentage of time for each path to be the better path in the path pair every hour. It is clear that there are a lot of CLR ranking changes each hour. And each path has some chance of being the better path in a path pair. So there is a room for voice quality improvement through path switching.

Table IX. CLR ranking comparison.

Path pair		Trace-file								
		1	2	3	4	5	6	7	8	9
AB	Number of CLR ranking changes	368	238	286	352	336	320	335	194	221
	A better[1] (%)	27.18	19.38	21.40	31.51	23.84	19.07	22.44	20.71	29.60
	B better[2] (%)	18.20	7.33	9.88	9.57	8.63	12.21	11.55	14.27	12.35
	Same[3] (%)	54.62	73.29	68.72	58.92	67.53	68.72	66.01	65.02	58.05
AC	Number of CLR ranking changes	209	178	234	196	208	134	200	140	185
	A better[4] (%)	17.51	6.25	9.35	8.36	11.75	6.27	9.80	5.68	17.78
	C better[5] (%)	20.19	8.11	10.81	11.79	10.04	13.92	12.76	14.59	13.34
	Same[6] (%)	62.30	85.64	79.84	79.85	78.21	79.81	77.44	79.73	68.88

[1]Percentage of time Path A has better quality compared to Path B in each hour.

[2]Percentage of time Path B has better quality compared to Path A in each hour.

[3]Percentage of time Path A and Path B have the same quality in each hour.

[4]Percentage of time Path A has better quality compared to Path C in each hour.

[5]Percentage of time Path C has better quality compared to Path A in each hour.

[6]Percentage of time Path A and Path C have the same quality in each hour.

2. Ideal case path switching control results

In ideal case path switching control, it is assumed that all the delays and losses of the two (or all) paths are known ahead of time. The ideal case path switching control gives an upper bound of the amount of voice quality improvement that can be achieved using path switching method.

The path switching control is performed every 400 ms. At each time step the CLR of the two paths in a path pair are calculated, and the VoIP packets are transmitted over the path with the least CLR. Table X shows the resulting CLR each hour (each trace-file has one hour data) of transmitting VoIP packets over only Path A and transmitting VoIP packets over only Path B without path switching, the resulting CLR using ideal case path switching control on path pair AB, and the percentage improvements of ideal case path switching control over no switching methods for path pair AB. The percentage improvement of method X over method Y is calculated as:

$$\frac{\text{Result of method X} - \text{Result of method Y}}{\text{Result of method Y}} \times 100\%. \tag{6.21}$$

So in this case, method X is the ideal case path switching control and method Y is one of these no switching methods. The resulting CLR each hour of transmitting over only Path C without path switching, the resulting ideal case path switching control on path pair AC, and the percentage improvements of ideal case path switching control over no switching methods for path pair AC, are also given in Table X.

As can be seen, for path pair AB, by using the ideal case path switching control the CLR is considerably reduced, from 10.89% on average, which indicates very bad voice quality, if the VoIP packets are sent only through Path A, and 20.39% on average, which is almost useless for VoIP application, if the VoIP packets are sent only through Path B, to 2.21% on average, which indicates an acceptable voice quality. For

path pair AC, by using the ideal case path switching control the CLR is considerably reduced, from 10.89% on average if the VoIP packets are sent only through Path A, and 8.94% on average, which is very bad for VoIP applications, if those packets are sent only through Path C, to 1.02% on average, which indicates a good voice quality. The VoIP QoS is significantly improved by using ideal case path switching control. For the path pair AB, it is 80.01% and 89.30% improvement in terms of CLR over transmitting VoIP packets only over Path A and only over Path B, respectively. For the path pair AC, it is 91.52% and 88.19% improvement in terms of CLR over transmitting VoIP packets only over Path A and only over Path C, respectively. However in reality, it is not possible to know all the delays and losses of the paths ahead of time, thus the voice quality improvement achieved by using ideal case path switching control is an upper bound of the improvement that can be achieved through path switching control.

The E-model MOS of the no switching methods results, the E-model MOS of the ideal case path switching control results, and the percentage improvement of the results of ideal case path switching control in terms of E-model MOS over the results of no switching methods in terms of E-model MOS are given in Table XI. It shows that for path pair AB, the resulting E-model MOS increases from 1.82 if the VoIP packets are only transmitted over Path A and 1.19 if the VoIP packets are only transmitted over Path B to 3.28 if the VoIP packets are transmitted using the idea case path switching control, which means the voice quality increases from unacceptable to fair. In terms of the resulting E-model MOS, using ideal case path switching control give 90% improvement over transmitting VoIP packets only over Path A, and 180% improvement over transmitting VoIP packets only over Path B. For path pair AC, the resulting E-model MOS increases from 1.82 if the VoIP packets are transmitted over Path A and 2.19 if the VoIP packets are transmitted over Path C to 3.52 if the VoIP

Table X. Ideal case path switching control results in terms of CLR.

Path pair		Trace-file									Avg.[1]	SD[2]
		1	2	3	4	5	6	7	8	9		
AB	Method	CLR results of path switching (%)										
	A[3]	18.53	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	3.83
	B[4]	25.21	15.59	18.54	26.23	19.53	15.75	18.83	18.46	25.35	20.39	4.14
	Ideal[5]	4.40	1.08	1.42	2.41	1.62	1.95	1.98	2.79	2.23	2.21	0.97
		Improvement of ideal switching over no switching (CLR)[6] (%)										
	A	76.25	83.15	83.01	73.66	78.40	81.48	80.94	81.12	81.99	80.00	3.24
	B	82.55	93.07	92.34	90.81	91.71	87.62	89.48	84.89	91.20	89.30	3.59
AC	Method	CLR results of path switching (%)										
	A	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	3.83
	C[7]	17.45	4.76	7.37	7.14	9.42	5.02	8.66	6.15	14.50	8.94	4.33
	Ideal	2.44	0.33	0.60	0.68	0.46	0.40	1.03	2.18	1.09	1.02	0.78
		Improvement of ideal switching over no switching (CLR)(%)										
	A	86.84	94.85	92.82	92.57	93.87	96.20	90.09	85.25	91.20	91.52	3.62
	C	86.02	93.07	91.86	90.48	95.12	92.03	88.11	64.55	92.48	88.19	9.27

[1]Average.

[2]Standard deviation.

[3]Transmitting the VoIP packets over Path A only, no switching.

[4]Transmitting the VoIP packets over Path B only, no switching.

[5]Ideal case path switching control.

[6]Percentage improvement of the results of ideal case path switching control in terms of CLR over the results of no switching methods in terms of CLR.

[7]Transmitting the VoIP packets over Path C only, no switching.

Table XI. Ideal case path switching results in terms of E-model MOS.

Path pair		Trace-file									Avg.	SD
		1	2	3	4	5	6	7	8	9		
AB	Method	MOS results of path switching (%)										
	A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	0.47
	B	0.99	1.43	1.19	0.99	1.19	1.46	1.22	1.26	1.00	1.19	0.18
	Ideal	2.88	3.51	3.44	3.24	3.40	3.33	3.32	3.17	3.28	3.28	0.18
		Improvement of Ideal switching over no switching (MOS)[1] (%)										
	A(%)	161	43	59	60	48	91	77	156	119	90	45
	B(%)	191	145	189	227	186	128	172	152	228	180	35
AC	Method	MOS results of path switching (%)										
	A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	0.47
	C	1.10	2.82	2.36	2.41	2.03	2.77	2.10	2.56	1.52	2.19	0.57
	Ideal	3.24	3.67	3.61	3.59	3.64	3.65	3.52	3.28	3.51	3.52	0.16
		Improvement of Ideal switching over no switching (MOS)(%)										
	A(%)	195	49	66	78	59	110	87	165	134	105	50
	C(%)	195	30	53	49	79	32	68	28	131	74	56

[1]Percentage improvement of the results of ideal case path switching control in terms of E-model MOS over the results of no switching methods in terms of MOS.

packets are transmitted using idea case path switching control, which means the voice quality increases from barely acceptable to almost good. In terms of the resulting E-model MOS, using ideal case path switching control gives 104% improvement over transmitting VoIP packets only over Path A, and 74% improvement over transmitting VoIP packets only over Path C. The ideal case results proves that potentially path switching control can significantly improve the VoIP QoS.

3. Simple predictor based path switching control results

Now consider a more realistic scenario. The path switching is performed very 400 ms. The two (or all) possible paths are probed every 100 ms. The network's condition in the last 400 ms is unknown due to the feedback delay of the probing packets. Only the network measurements more than 400 ms ago are available. Then in order to make a path switching decision at the current time step, the network's condition for next 400 ms should be predicted based on the available network measurements of each path. The simplest way is to use a two-step-ahead simple predictor (SP) for prediction.

Without losing generality, assume that the path switching control is performed using paths A and B, then the two-step-ahead simple predictors for Path A and Path B are given by

$$\begin{aligned} \hat{y}_A(k) &= y_A(k-2) \\ \hat{y}_B(k) &= y_B(k-2) \end{aligned}, \tag{6.22}$$

where k denotes the current time step, $\hat{y}_A(k)$ and $\hat{y}_B(k)$ are the predicted CLRs at the current time step for paths A and B, and $y_A(k-2)$ and $y_B(k-2)$ are the measurements of theses two CLRs. Then, $\hat{y}_A(k)$ and $\hat{y}_B(k)$ are compared, and VoIP packets are transmitted over the path with the smaller CLR. If the predicted CLRs of the two paths are the same, then the path chosen in the previous time step is used.

The resulting CLR of SP based path switching control are presented in Table XII. Although the improvement in the results of SP based path switching control over no switching methods in terms of CLR is not as much as the improvement in the results of ideal case path switching control over no switching methods in terms of CLR, it is still significant. The SP based path switching control reduces the resulting CLR of path pair AB to an average of 6.05%, an improvement of 41.07% and 69.90% over

transmitting VoIP packets only over Path A and only over Path B, respectively. The resulting CLR of path pair AC is reduced to an average of 3.26%, an improvement of 68.37% and 59.48% over transmitting only over Path A and and only over Path C, respectively. So SP based path switching control can realized about half of the possible improvement of ideal case path switching control in terms of the resulting CLR.

Fig. 28 shows a comparison of the no switching methods results, the ideal case path switching control results, and SP base path switching control results. The figure shows that ideal case path switching control can significantly reduce the resulting CLR thus improve the voice quality, and SP based path switching control can realize about half of this possible CLR improvement.

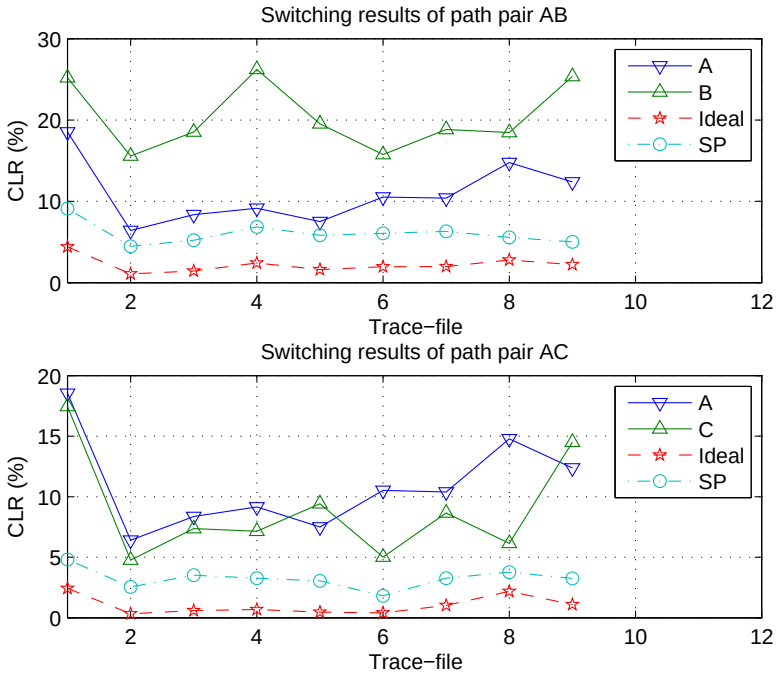

Fig. 28. No switching, ideal case and SP based path switching results in terms of CLR. Top: path switching results of path pair AB in terms of CLR. Bottom: path switching results of path pair AC in terms of CLR.

Table XII. Simple predictor based path switching results in terms of CLR.

Path pair		Trace-file									Avg.	SD
		1	2	3	4	5	6	7	8	9		
AB	Method	CLR results of path switching (%)										
	A(%)	18.53	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	3.83
	B(%)	25.21	15.59	18.54	26.23	19.53	15.75	18.83	18.46	25.35	20.39	4.14
	SP[1]	9.11	4.48	5.21	6.85	5.83	6.07	6.31	5.56	5.02	6.05	1.35
		Improvement of SP switching over no switching (CLR)[2] (%)										
	A(%)	50.84	30.11	37.68	25.14	22.27	42.36	39.27	62.38	59.45	41.05	14.27
	B(%)	63.86	71.26	71.90	73.88	70.15	61.46	66.49	69.88	80.20	69.90	5.56
AC	Method	CLR results of path switching (%)										
	A(%)	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	3.83
	C(%)	17.45	4.76	7.37	7.14	9.42	5.02	8.66	6.15	14.50	8.94	4.33
	SP	4.81	2.55	3.52	3.27	3.05	1.82	3.27	3.77	3.25	3.26	0.82
		Improvement of SP switching over no switching (CLR) (%)										
	A(%)	74.06	60.22	57.89	64.26	59.33	82.72	68.53	74.49	73.75	68.36	8.51
	C(%)	72.44	46.43	52.24	54.20	67.62	63.75	62.24	38.70	77.59	59.47	12.60

[1]SP based path switching control.

[2]Percentage improvement of the results of SP based path switching control in terms of CLR over the results of no switching methods in terms of CLR.

The E-model MOS of the SP based path switching control results are presented in Table XIII. The SP based path switching control improves the E-model MOS of path pair AB to an average of 2.60, an improvement of 51% and 121% over transmitting only over Path A and only over Path B, respectively. For path pair AC, the average resulting E-model MOS using SP based path switching control is 3.08, an improvement of 80% and 52% over transmitting only over Path A and only over Path B, respectively.

Fig. 29 shows a comparison of the no switching method results, the ideal case path switching control results, and the SP base path switching control results in terms of E-model MOS. The figure shows that ideal case path switching control significantly improves the voice quality in terms of E-model MOS, and SP based path switching control can realize about half of that possible improvement in terms of E-model MOS.

C. Chapter Summary

This chapter discusses the problems that appear in collecting trace-file data from an actual network. The solutions to over come these problems are given. By comparing the CLR ranking changes, it shows that there is a room for voice quality improvement using path switching control. Preliminary study using ideal case path switching control shows that the voice quality improvement by using path switching control could be significant. Preliminary study using the more realistic SP based two-steps-head predictive path switching control shows that a predictive path switching control, even when using a simple predictor, can realize about half of the possible voice quality improvement.

Table XIII. Simple predictor based path switching results in terms of E-model MOS.

Path pair		Trace-file									Avg.	SD
		1	2	3	4	5	6	7	8	9		
AB	Method	MOS results of path switching (%)										
	A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	0.47
	B	0.99	1.43	1.19	0.99	1.19	1.46	1.22	1.26	1.00	1.19	0.18
	SP	2.14	2.86	2.74	2.45	2.63	2.60	2.56	2.69	2.77	2.60	0.21
		Improvement of SP switching over no switching (MOS)[1] (%)										
	A (%)	95	16	26	21	15	49	36	117	85	51	38
	B (%)	116	100	130	147	121	78	110	113	177	121	28
AC	Method	MOS results of path switching (%)										
	A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	0.47
	C	1.10	2.82	2.36	2.41	2.03	2.77	2.10	2.56	1.52	2.19	0.57
	SP	2.81	3.21	3.03	3.08	3.12	3.36	3.08	2.98	3.08	3.08	0.15
		Improvement of SP switching over no switching (MOS) (%)										
	A (%)	155	30	40	52	36	93	64	140	105	80	46
	C (%)	155	14	28	28	54	21	47	16	103	52	48

[1]Percentage improvement of the results of SP based path switching control in terms of E-model MOS over the results of no switching methods in terms of E-model MOS.

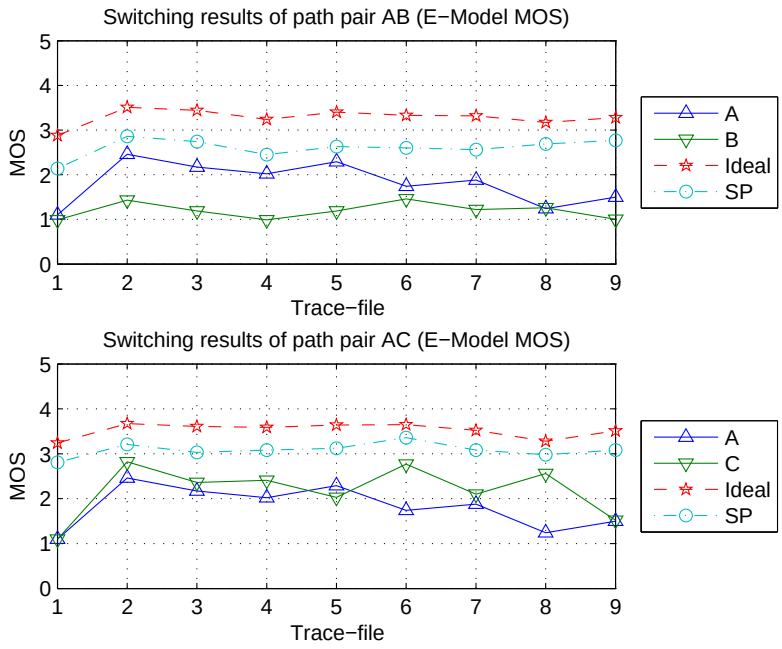

Fig. 29. No switching, ideal case and SP based path switching results in terms of E–model MOS. Top: path switching results of path pair AB in terms of E-model MOS. Bottom: path switching results of path pair AC in terms of E-model MOS.

CHAPTER VII

PREDICTOR DEVELOPMENT FOR PREDICTIVE PATH SWITCHING CONTROL USING ACTUAL NETWORK MEASUREMENTS

In this chapter, problems related to predictor development in predictive path switching control are addressed. The requirement placed on the prediction horizon is investigated. Topics, such as the minimum prediction horizon and the separation of switching decisions, are discussed. The relation between the switching interval and the control results is investigated. Finally, the results of predictor development are presented.

A. Requirements on the Prediction Horizon

1. Minimum prediction horizon

Given that there is an information feedback delay limit τ, at any given time t only the measurements before $t-\tau$ ago are available, because of the delay in the feedback of the probing packets. These measurements are either delay measurements obtained from echoed probing packets, or deduced losses because of the timeout in probing packets based on the information feedback delay limit. To perform predictive path switching control, predictions of future information of the path are needed. These predictions can only be obtained based on the available measurements. Thus, the minimum prediction horizon required is τ. Once the prediction step size T_p is decided, the required horizon can be expressed in terms of minimum required number of steps for prediction.

In this study, this information feedback delay limit τ is 400 ms. That is, it is assumed that it takes at most 400 ms for information feedback to reach the sender on

a particular packet. Define $y(k)$ as the information signal of the step k, which is an average of the path condition from the time kT_p to time $(k+1)T_p$. If the prediction step size T_p is 400 ms or more, then at the beginning of step k, i.e. time kT_p, the information signal $y(k-1)$ of step $k-1$ is not available. Because obtaining $y(k-1)$ needs the measurements from time $kT_p - 400$ ms to time kT_p, which is not available at time kT_p. Only values of the signal $y(k-2)$ of step $k-2$ and before are available for use. A predictor that can provide at least two-step-ahead prediction will be required to predict the future information signal, i.e. $\hat{y}(k|k-2)$. If the prediction step size T_p is less than τ, then more number of prediction steps are required. For example, if the prediction step size is 100 ms, then the latest information available is $y(k-5)$, which contains the information from the past 500 ms to past 400 ms. A predictor that can provide at least a five-step-ahead prediction is required. These two cases are illustrated in Fig. 30. The minimum number of required prediction steps for a prediction step size of T_p is given by

$$P = \left\lceil \frac{\tau}{T_p} \right\rceil + 1. \tag{7.1}$$

2. Separation of switching decisions

In a typical Model Predictive Control (MPC) set up, the effectiveness of the control depends on two factors: the accuracy of the predictions and the number of prediction steps into the future that are used. Normally, accurate predictions and large number of prediction steps into the future give better control results. Thus larger prediction horizons result in better control. But that is not the case for the predictive path switching control in this study. There is a *separation of switching decisions* in this study, which can be stated as: *in predictive path switching control as this study, a control action taken at one time will not interference with any future control actions.*

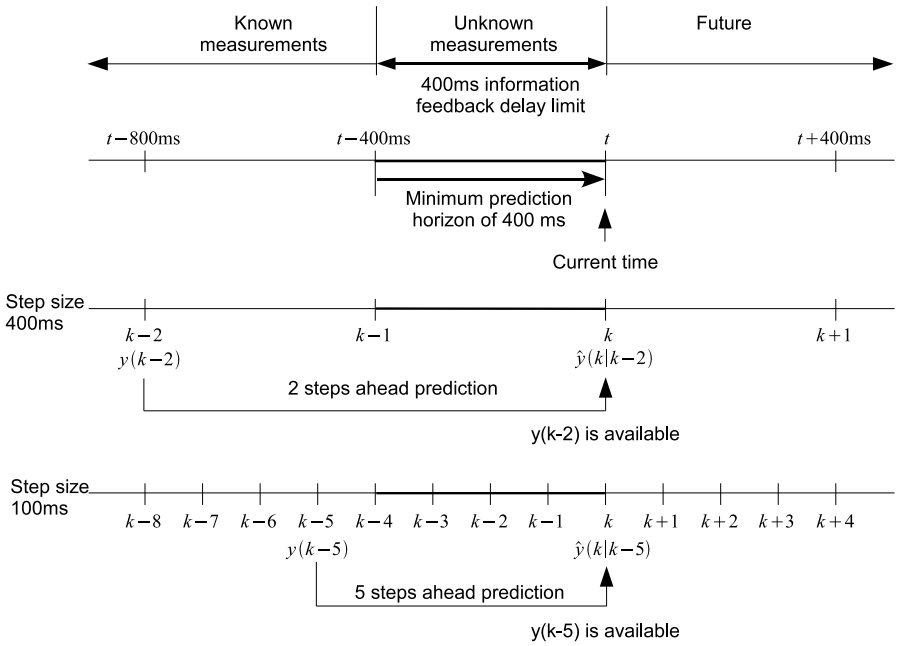

Fig. 30. Minimum prediction horizon and number of steps ahead.

Each control action can be determined independently. It is caused by the underlying assumption of this research.

Without losing generality, take the case of predictive switching between Path A and Path B for a VoIP flow, shown in Fig. 31. The flow sent is marked as $F_{in}(t)$. The two paths have their own CLRs, which are the probability of a packet being lost or over-delayed when sending through a given path. Under the assumption that the VoIP flow of interest is a relatively small amount of the flow compared to all the other flows in the network, these CLRs are determined by the cross traffic in the networks. The VoIP flow of interest has insignificant impact on these CLRs. Assume that the CLRs of the two paths at time t are $y_A(t)$ and $y_B(t)$, respectively. Because of the assumption made, $y_A(t)$ and $y_B(t)$ are not affected by the control actions applied to the VoIP flow of interest. When $y_A(t)$ and $y_B(t)$ are discretized, they can be viewed

as time series.

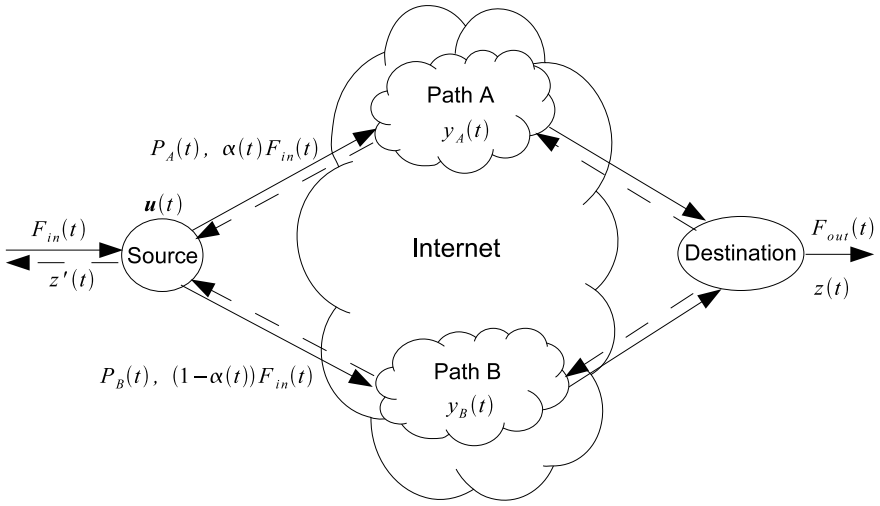

Fig. 31. Sketch of predictive path switching control.

To measure the CLRs, probing packets are sent every T_s through the two paths. The probing flows are marked as $P_A(t)$ and $P_B(t)$. The paths CLRs are predicted every T_p. Based on these predictions, a sequence of path switchings are determined every T_d. This sequence of path switchings are applied one by one every T_u. T_p should be integer multiples of T_s, T_u should be integer multiples of T_p, and T_d should be integer multiples of T_u.

Define the the control signal $u(t)$ at time t as a decision on the fraction of the flow to transmit over Path A and on the fraction to transmit over Path B. Designated the proportion transmitted over path A at time t as $\alpha(t)$, then

$$\mathbf{u}(t) = \begin{bmatrix} \alpha(t) \\ 1 - \alpha(t) \end{bmatrix}. \tag{7.2}$$

It is clear that $\mathbf{u}(t)$ has the constraints as follows:

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{u}(t) = \alpha(t) + [1 - \alpha(t)] = 1, \tag{7.3}$$

and

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} \alpha(t) \\ 1 - \alpha(t) \end{bmatrix} = \mathbf{u}(t) \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{7.4}$$

The output flow at the destination is marked as $F_{out}(t)$. The CLR experienced by this flow is calculated at the destination and is designated as $z(t)$. This result, $z(t)$, is fed back to the source and designated as $z'(t)$. The CLR experienced by the output flow is a delayed function of the control action and paths CLRs,

$$z(t) = \alpha(t - \tau_f) y_A(t - \tau_f) + (1 - \alpha(t - \tau_f)) y_B(t - \tau_f), \tag{7.5}$$

where, τ_f is the forward delay limit. When $z(t)$ is fed back to the source, the system model becomes

$$\begin{aligned} z'(t) &= z(t - \tau_r - \tau_b) \\ &= \alpha(t - \tau) y_A(t - \tau) + (1 - \alpha(t - \tau)) y_B(t - \tau) \\ &= \begin{bmatrix} y_A(t - \tau) & y_B(t - \tau) \end{bmatrix} \mathbf{u}(k - \tau), \end{aligned} \tag{7.6}$$

where, τ_r is the wait time of the information at the destination before is sent to the source, τ_b is the backward delay, and $\tau = \tau_f + \tau_r + tau_b$ is the information feedback delay. These results show that any control action taken at time t will impact the measurement of the CLR experienced by the output flow at the destination at time $t + \tau_f$, and the measurement is obtained at the source at time $t + \tau$. Define

$$\mathbf{y}(t - \tau) = \begin{bmatrix} y_A(t - \tau) \\ y_B(t - \tau) \end{bmatrix}, \tag{7.7}$$

then

$$z'(t) = \mathbf{y}^T(t-\tau)\mathbf{u}(k-\tau). \tag{7.8}$$

If T_p and T_d are set equal to T_u, then equation (7.6) becomes

$$\begin{aligned} z'(k) &= \alpha(k-d)y_A(k-d) + (1-\alpha(k-d))y_B(k-d), \\ &= \begin{bmatrix} y_A(k-d) & y_B(k-d) \end{bmatrix} \mathbf{u}(k-d) \\ &= \mathbf{y}^T(k-d)\mathbf{u}(k-d), \end{aligned} \tag{7.9}$$

where,

$$d = \left\lceil \frac{\tau}{T_u} \right\rceil, \tag{7.10}$$

with $\lceil x \rceil$ designating the ceiling function which gives the smallest integer greater or equal to x, is the dead-time of the system. The prediction of future system outputs beyond the dead-time d and up to the control horizon p are

$$\begin{bmatrix} \hat{z}'(k+d) \\ \hat{z}'(k+d+1) \\ \cdots \\ \hat{z}'(k+p) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}}^T(k) & & & 0 \\ & \hat{\mathbf{y}}^T(k+1) & & \\ & & \ddots & \\ & 0 & & \hat{\mathbf{y}}^T(k+p-d) \end{bmatrix} \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+p-d) \end{bmatrix}. \tag{7.11}$$

If the desired outputs are zero CLRs, the future control sequence

$$[\mathbf{u}^T(k) \ \cdots \ \mathbf{u}^T(k+p-d)]^T$$

is calculated by minimizing the cost function

$$\begin{aligned} J(k) &= \sum_{i=0}^{p-d} \Big[(\hat{z'}(k+d+i) - 0)^2 + \lambda \mathbf{u}^T(k+i)\mathbf{u}(k+i) \Big] \\ &= \sum_{i=0}^{p-d} \big[(\hat{\mathbf{y}}(k+i)\mathbf{u}(k+i))^2 + \lambda \mathbf{u}^T(k+i)\mathbf{u}(k+i) \big] \\ &= \sum_{i=0}^{p-d} \big[\mathbf{u}^T(k+i)(\hat{\mathbf{y}}^T(k+i)\hat{\mathbf{y}}(k+i) + \lambda)\mathbf{u}(k+i) \big], \end{aligned} \tag{7.12}$$

under the constraints that

$$\left[\begin{array}{cc|cc|ccc|cc} 1 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 1 & 1 \end{array}\right] \left[\begin{array}{c} \mathbf{u}(k) \\ \hline \mathbf{u}(k+1) \\ \hline \vdots \\ \hline \mathbf{u}(k+p) \end{array}\right] = \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \end{array}\right], \tag{7.13}$$

$$\left[\begin{array}{c} 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline \vdots \\ \hline 0 \\ 0 \end{array}\right] \leq \left[\begin{array}{c} \mathbf{u}(k) \\ \hline \mathbf{u}(k+1) \\ \hline \vdots \\ \hline \mathbf{u}(k+p) \end{array}\right] \leq \left[\begin{array}{c} 1 \\ 1 \\ \hline 1 \\ 1 \\ \hline \vdots \\ \hline 1 \\ 1 \end{array}\right]. \tag{7.14}$$

Here λ is the weight used to penalize the control actions. In this study, this λ is set to 0, which means that there is no cost penalty for the control actions. When there is no penalty for control actions, for the optimal control results, the proportion of VoIP flow transmitted over path A at time t ($\alpha(t)$) is either 0 (none) or 1 (all). Because the path CLRs $\{\mathbf{y}(i)\}$ are independent of the control actions $\{\mathbf{u}(i)\}$, each $\hat{z'}^2(k+d+i)$ term in the cost function (7.12) can be minimized independently by

the corresponding $\mathbf{u}(k+i)$ within its constraints. So to determine the control action $\mathbf{u}(k)$ at time k that minimizes the cost function $J(k)$, only the prediction for $\mathbf{y}(k)$ is needed. This prediction of $\mathbf{y}(k)$ can be obtained using a $P = d + 1$ steps ahead predictor based on the available measurements $\{\mathbf{y}(k-d-n)|n = 1, 2, 3 \ldots\}$. There is no need for predicting more steps ahead.

3. Section summary

In the section the requirement for the prediction horizon is discussed. The relation between the prediction step size, control step size, decision step size, and the minimum required prediction horizon is discussed. It is demonstrated that if the prediction step size, control step size, and decision step size are all set to 400 ms, a prediction horizon of two-step-ahead and only two-step-ahead is needed for the predictive path switching control used in this study.

B. The Impact of Switching Interval

1. Switching interval investigation method

In the following studies the prediction step size T_p, control step size T_u, and decision step size T_d are all set to a same value. In this section, the impact of this prediction step size T_p, which is the same as the control switching interval T_u, is investigated. The T_p values of 100 ms, 200 ms, 400 ms, and 800 ms are investigated. According to equation (7.1), these intervals will require 5-step-ahead, 3-step-ahead, 2-step-ahead, and 2-step-ahead predictions, respectively. In this study the probing interval T_s is 100 ms. The fastest switching is packet to packet switching. It gives the shortest switching interval of 100 ms. The information feedback delay limit is 400 ms. Large switching intervals can be more than this 400 ms. The considered switching intervals

cover this whole range of switching intervals.

Three sets of data are considered as three separate paths, Path A, Path B, and Path C. Each data set has 9 one hour long trace-files. These data sets are obtained on different days by actively probing between the *planetlab1.nbgisp.com* node and the *planetlab1.gti-dsl.nodes.planet-lab.org* node on PlanetLab. The average CLRs of Path A, Path B, and Path C are 11%, 20%, and 9%, respectively. Two groups of studies are conducted. One for switching between Path A and Path B. And one for switching between Path A and Path C. The CLRs, $\{y_{i,T_p}(k)|i = A, B, C\}$, of each path are calculated for every switching interval T_p, and are used as the information signal for path switching control.

Three different types of predictive path switching controls, ideal case path switching control, simple predictor (SP) based path switching control, and autoregressive predictor (AR) based path switching control, are attempted with these prediction/switching intervals. For the ideal case path switching control, all future values of the information signal are assumed known. The $y_{i,T_p}(k)$ $(i = A, B, C)$ are used directly in calculating the control action $u_{T_p,ideal}(k)$. For SP based path switching control, the P steps ahead prediction is given by

$$\hat{y}_{i,T_p,SP}(k|k-P) = y_{i,T_p}(k-P). \tag{7.15}$$

The $\hat{y}_{i,T_p,SP}(k)$ $(i = A, B, C)$ are used in calculating the control action $u_{T_p,SP}(k)$. For AR based path switching control, the CLRs $\{y_{i,T_p}(k)|i = A, B, C\}$ of a path are treated as a time series, and its P steps ahead AR predictor is given by

$$\begin{aligned} \hat{y}_{i,T_p,AR}(k|k-P) &= a_1 y_{i,T_p}(k-P) + a_2 y_{i,T_p}(k-P-1) \\ &\quad + \ldots + a_{n_a} y_{i,T_p}(k-P-n_a+1), \end{aligned} \tag{7.16}$$

where, n_a is the predictor order. The $\hat{y}_{i,T_p,AR}(k|k-P)$ $(i = A, B, C)$ are used in calculating the control action $u_{T_p,AR}(k)$. The AR model of a path's CLR signal is developed by using the first trace-file of that path. The first half of the data is used for training, the second half of the data is used for testing. Mean square error (MSE)

$$MSE_{AR} = \frac{\sum_{k=1}^{N} \left(\hat{y}_{i,T_p,AR}(k|k-P) - y_i(k)\right)^2}{\sum_{k=1}^{N} y_i^2(k)} \tag{7.17}$$

is used for testing. The model with the best model order n_a, which gives the minimum mean square error (MSE) on the test set, is chosen.

SP based path switching control and AR based path switching control are more realistic than the ideal case path switching control. Because SP based path switching control and AR based path switching control take the delay of the information feedback into consideration.

2. Switching interval investigation results

The resulting CLRs of the VoIP flow switching between Path A and Path B using different switching intervals and different path switching control strategies are presented in Table XIV. The resulting CLRs are also plotted in Fig. 32. The resulting CLRs of switching between Path A and Path C are presented in Table XIV, and are plotted in Fig. 33.

From the results of the ideal case path switching control between Path A and Path B, and the results of the ideal case path switching control between Path A and Path C, it can be observed that smaller switching interval gives lesser (better) resulting CLRs. The difference is around 0.1 percentage points in terms of CLR when switching between Path A and Path B. The difference is below 0.1 percentage points when switching between Path A and Path C. These results indicate that for the ideal case path switching control, the impact of switching interval is not very significant.

Table XIV. Resulting CLRs of different switching intervals.

Path pair	Method	Interval[1](ms)	CLRs of different switching intervals on trace-file (%)									Avg[2]. (%)	Std[3]. (%)
			1	2	3	4	5	6	7	8	9		
AB	Ideal[4]	100	4.29	1.02	1.34	2.30	1.53	1.83	1.86	2.73	2.17	2.12	0.96
		200	4.32	1.04	1.37	2.34	1.55	1.86	1.90	2.76	2.18	2.15	0.97
		400	4.40	1.08	1.42	2.41	1.62	1.95	1.98	2.79	2.23	2.21	0.97
		800	4.62	1.18	1.58	2.57	1.78	2.14	2.18	2.89	2.37	2.37	0.99
	SP[5]	100	8.37	3.87	4.57	6.19	5.29	5.46	5.61	5.09	4.62	5.45	1.29
		200	8.61	4.09	4.80	6.45	5.47	5.65	5.90	5.26	4.78	5.67	1.30
		400	9.11	4.48	5.21	6.85	5.83	6.07	6.31	5.56	5.02	6.05	1.35
		800	11.04	5.89	6.75	9.04	7.69	7.80	8.01	6.84	6.16	7.69	1.59
	AR[6]	100	8.18	3.41	4.54	5.94	5.12	5.56	5.62	5.40	4.52	5.37	1.31
		200	8.64	3.62	4.32	6.16	5.45	5.49	5.69	5.68	4.51	5.51	1.43
		400	9.18	3.83	5.02	6.37	5.93	5.75	6.03	5.98	4.53	5.85	1.50
		800	11.10	4.98	5.98	7.87	6.45	7.03	7.20	7.16	5.38	7.02	1.79
AC	Ideal	100	2.39	0.31	0.57	0.65	0.44	0.37	1.01	2.15	1.04	0.99	0.77
		200	2.39	0.32	0.58	0.66	0.45	0.38	1.01	2.15	1.06	1.00	0.77
		400	2.44	0.33	0.60	0.68	0.46	0.40	1.03	2.18	1.09	1.02	0.78
		800	2.54	0.39	0.70	0.72	0.51	0.45	1.07	2.20	1.16	1.08	0.78
	SP	100	4.51	2.23	2.99	2.95	2.72	1.65	2.92	3.50	2.94	2.93	0.79
		200	4.60	2.37	3.16	3.07	2.84	1.74	3.05	3.60	3.02	3.05	0.79
		400	4.81	2.55	3.52	3.27	3.05	1.82	3.27	3.77	3.25	3.26	0.82
		800	5.84	3.59	4.73	4.44	4.21	2.44	4.14	4.47	4.03	4.21	0.91
	AR	100	4.74	2.24	3.05	2.96	2.75	1.68	2.96	3.57	3.10	3.01	0.85
		200	4.95	2.37	3.24	2.98	2.89	1.77	3.03	3.67	3.28	3.13	0.88
		400	5.25	2.52	3.56	3.31	3.20	2.16	3.16	3.78	3.25	3.35	0.87
		800	5.97	3.56	4.82	4.54	4.46	2.84	4.12	4.63	4.17	4.35	0.86

[1]Prediction/switching interval.

[2]Average.

[3]Standard deviation.

[4]Ideal case path switching control.

[5]SP based path switching control.

[6]AR based path switching control.

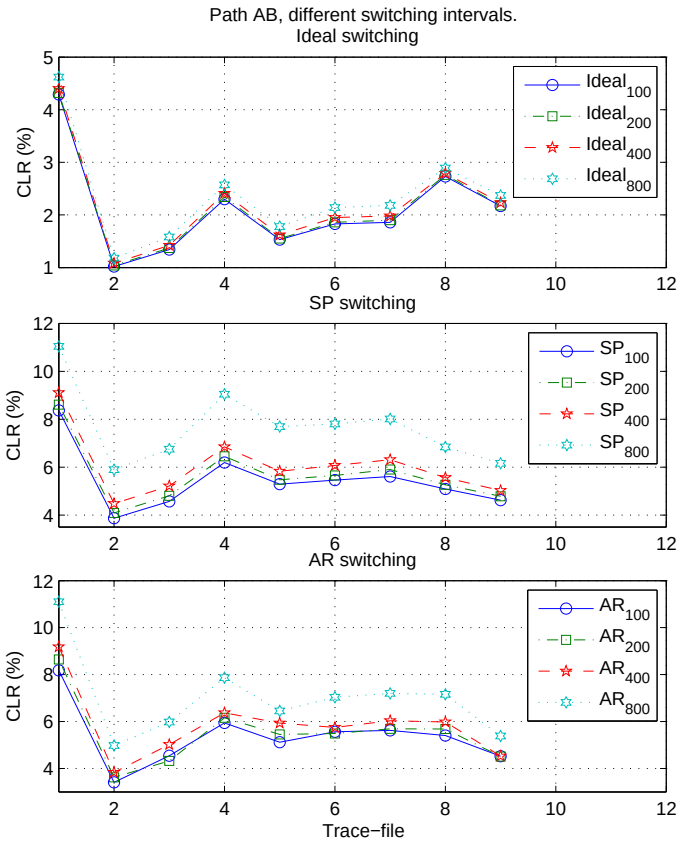

Fig. 32. CLR results of path switching between Path A and Path B with different intervals. Top: Ideal case path switching control. Middle: SP based path switching control. Bottom: AR based path switching control.

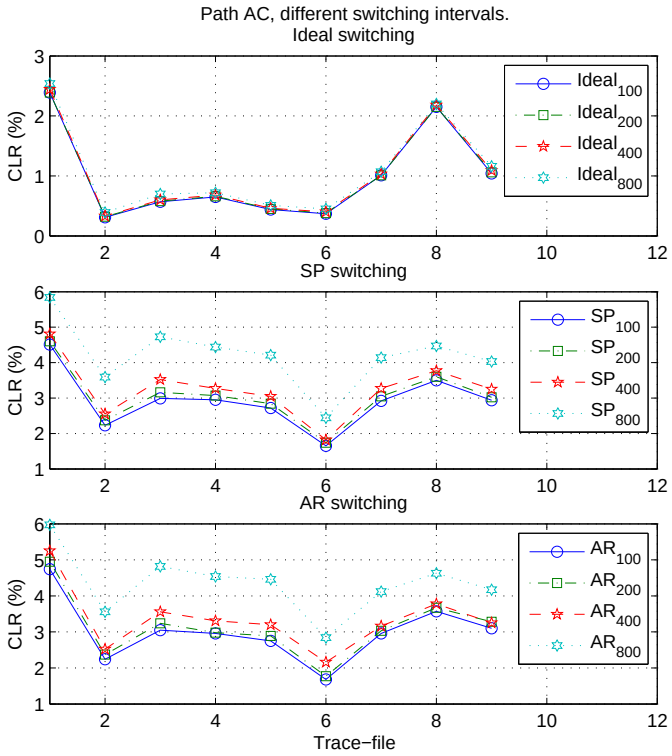

Fig. 33. CLR results of path switching between Path A and Path C with different intervals. Top: Ideal case path switching control. Middle: SP based path switching control. Bottom: AR based path switching control.

Generally the prediction accuracy deteriorates as the number of prediction steps increases [154]. Smaller prediction/switching step size requires more number of prediction steps to overcome the information feedback delay limit. But the results of SP based path switching control and AR based path switching control show that smaller prediction/switching step size still gives lesser CLRs than larger prediction/switching step size. This result indicates that faster switching is better.

The 100 ms, 200 ms, and 400 ms switching intervals are all within the information feedback delay limit of 400 ms. In both SP based path switching control and AR based path switching control, the resulting CLRs of the 100 ms, 200 ms, and 400 ms switching intervals are all fairly close to each other. They have differences of less than 0.5 percentage points. The 800 ms switching interval is more than the information feedback delay limit of 400 ms. In both SP based path switching control and AR based path switching control, the resulting CLRs of the 800 ms switching interval are more than 1.0 percentage points worse than the resulting CLRs of the faster switching intervals.

The voice quality of different switching intervals and different path switching controls are presented in Table XV in terms of E-model MOS. The resulting E-model MOSs of path pair AB are plotted in Fig. 34. The resulting E-model MOSs of path pair AC are plotted in Fig. 35.

The same trend as shown in the CLR results can be observed in the E-model MOS results as well. For the ideal case path switching control, smaller switching interval gives better resulting MOSs. Yet the difference is less than 0.05 which is not very significant. For SP based path switching control and AR based path switching control, smaller prediction/switching step size also gives better MOSs. The results indicate that faster switching is better. In both SP based path switching control and AR based path switching control, the differences in the resulting E-model MOSs of

Table XV. Resulting E-model MOS of different switching intervals.

Path pair	Method	Interval(ms)	E-model MOS of different switching intervals on trace-file									Avg.	SD
			1	2	3	4	5	6	7	8	9		
AB	Ideal	100	2.90	3.52	3.46	3.26	3.42	3.36	3.35	3.18	3.29	3.30	0.18
		200	2.89	3.52	3.45	3.25	3.41	3.35	3.34	3.18	3.29	3.30	0.18
		400	2.88	3.51	3.44	3.24	3.40	3.33	3.32	3.17	3.28	3.28	0.18
		800	2.84	3.49	3.40	3.21	3.36	3.29	3.28	3.15	3.25	3.25	0.19
	SP	100	2.24	2.97	2.85	2.55	2.72	2.70	2.68	2.76	2.84	2.70	0.21
		200	2.21	2.93	2.81	2.51	2.69	2.67	2.63	2.74	2.81	2.67	0.21
		400	2.14	2.86	2.74	2.45	2.63	2.60	2.56	2.69	2.77	2.60	0.21
		800	1.89	2.61	2.48	2.12	2.33	2.32	2.29	2.47	2.57	2.34	0.23
	AR	100	2.27	3.05	2.85	2.59	2.75	2.69	2.68	2.71	2.85	2.72	0.21
		200	2.20	3.01	2.89	2.55	2.69	2.69	2.66	2.67	2.86	2.69	0.23
		400	2.13	2.97	2.77	2.52	2.61	2.65	2.61	2.62	2.85	2.64	0.24
		800	1.88	2.76	2.59	2.28	2.51	2.43	2.40	2.43	2.69	2.44	0.26
AC	Ideal	100	3.25	3.68	3.62	3.60	3.65	3.66	3.52	3.29	3.52	3.53	0.16
		200	3.24	3.67	3.62	3.60	3.64	3.66	3.52	3.29	3.51	3.53	0.16
		400	3.24	3.67	3.61	3.59	3.64	3.65	3.52	3.28	3.51	3.52	0.16
		800	3.22	3.66	3.59	3.59	3.63	3.64	3.51	3.28	3.49	3.51	0.16
	SP	100	2.86	3.27	3.13	3.14	3.18	3.39	3.14	3.03	3.14	3.14	0.15
		200	2.84	3.25	3.10	3.11	3.16	3.37	3.12	3.01	3.12	3.12	0.15
		400	2.81	3.21	3.03	3.08	3.12	3.36	3.08	2.98	3.08	3.08	0.15
		800	2.63	3.02	2.82	2.87	2.91	3.23	2.92	2.86	2.94	2.91	0.16
	AR	100	2.82	3.27	3.12	3.14	3.17	3.38	3.13	3.02	3.11	3.13	0.16
		200	2.78	3.25	3.08	3.13	3.15	3.37	3.12	3.00	3.07	3.11	0.16
		400	2.74	3.22	3.02	3.07	3.09	3.29	3.10	2.98	3.08	3.06	0.15
		800	2.60	3.02	2.80	2.85	2.87	3.16	2.92	2.83	2.91	2.89	0.15

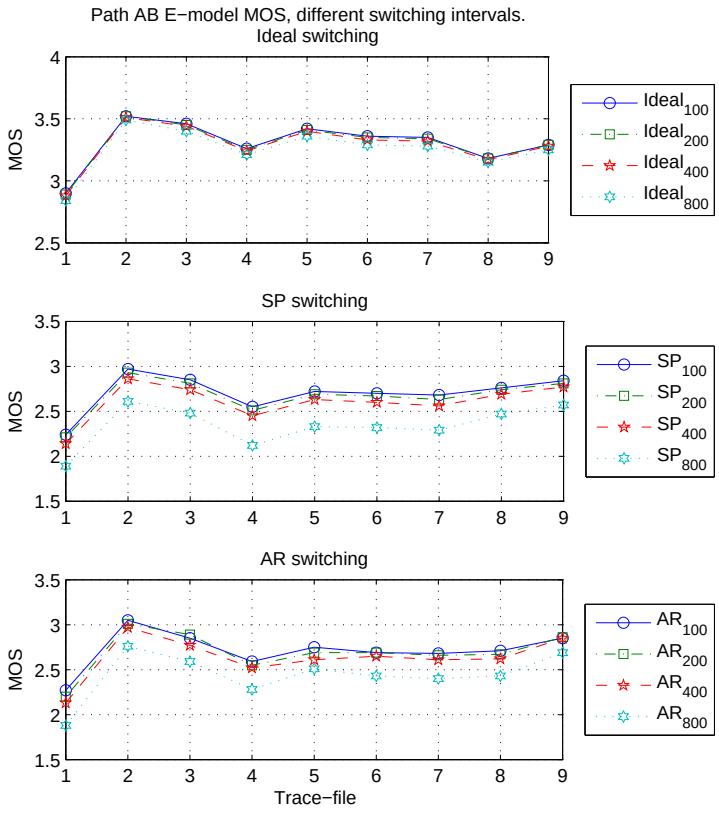

Fig. 34. E-model MOS results of path switching between Path A and Path B with different intervals. Top: Ideal case path switching control. Middle: SP based path switching control. Bottom: AR based path switching control.

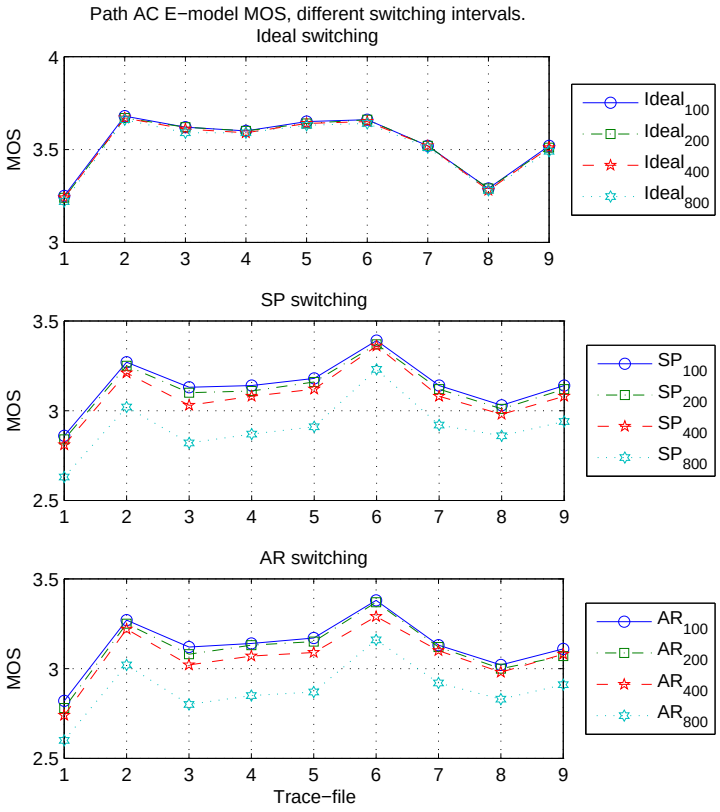

Fig. 35. E-model MOS results of path switching between Path A and Path C with different intervals. Top: Ideal case path switching control. Middle: SP based path switching control. Bottom: AR based path switching control.

the 100 ms, 200 ms, and 400 ms switching intervals are less than 0.1. Where as the resulting E-model MOS of the 800 ms switching interval are more than 0.1 worse than the resulting E-model MOSs of the other switching intervals.

3. Section summary

In this section, the impact of prediction/switching step size (interval) T_p on the predictive path switching control results is investigated. The results show that smaller switching interval gives better switching results. Yet, when the switching intervals are less than the information feedback delay limit, the differences are pretty small. On the other hand the switching results deteriorate a lot when the switching interval is more than the information feedback delay limit. For further studies in this research, the prediction/switching interval T_p is set equal to the information feedback delay limit of 400 ms. This is because all the path switching control results with switching interval below 400 ms are pretty close, while a slower switching interval will considerably reduce the demand of computation power.

C. Predictor Development

In this section, system identification techniques are used to develop two-step-ahead predictors. Fig. 36 shows the system identification loop given by Ljung [155]: first collect data, then choose a model set, and finally pick the "best" model in this set. The objectives of this section are to find the best information signals to use for the predictive path switching control, to decide which type of predictor is the best, and to figure out the best parameters for the predictors.

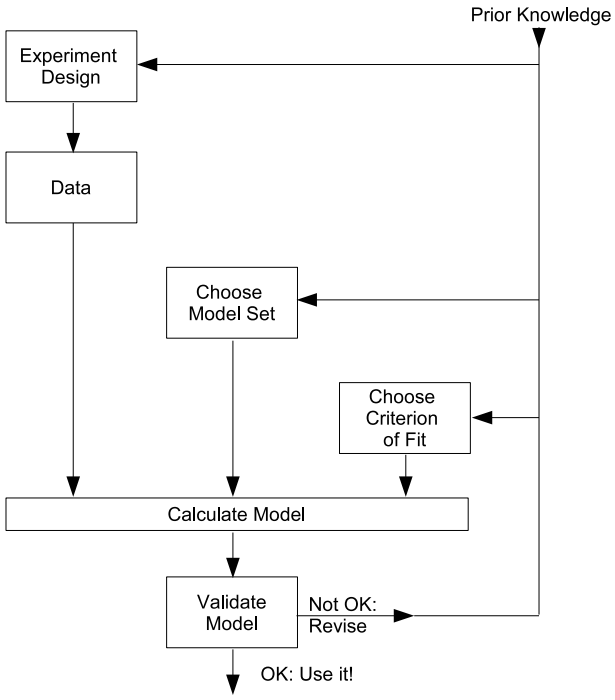

Fig. 36. The system identification loop [155].

1. Information signals used in prediction

The trace-files of the paths are obtained through actively probing the network with probing packets. These probing packets are sent at the sampling interval T_s of 100 ms. The probing packets are 38 bytes UDP packets. They are sent by the sender side program at the send-time t_{send} and are received by the receiver side program at the receive-time t_{recv}. The receiver side program echo these packets back to the sender side by piggybacking its own probing packets at the response-time t_{resp}. The echoed packets are received at the send side at the arrive-time t_{arrv}. The sequence number SN, the send-time t_{send}, and the arrive-time t_{arrv} of these packets are recorded at the sender side. The sequence number SN, the receive-time t_{recv}, and the response-time t_{resp} of these packets are recorded at the receiver side. From these measurements, the delays and losses of the paths can be obtained. Then, the following three types of information signals are calculated, and are used in path quality prediction for predictive path switching control.

a. Comprehensive loss rate

The first type of information signal used for comparing path quality is the *comprehensive loss rate (CLR)*. CLR of a path at time t is defined as the the chance of a packet being lost or over-delayed when sending through that path at that time. In this study, a packet with a delay over 150 ms is defined as an over-delayed packet. The CLR of an interval is calculated as the ratio of the sum of the number of lost packets (N_{loss}) and the number of the over-delayed packets ($N_{overdelay}$) to the total number of packets sent (N_{total}) in that interval.

$$CLR = \frac{N_{loss} + N_{overdelay}}{N_{total}} \times 100\%. \tag{7.18}$$

For real-time multimedia applications such as VoIP, an over-delayed packet has the same negative impact as a lost packet to the conversation quality. The CLR signal takes into account both the effect of the lost packets and the effect of over-delayed packets. As a rule of thumb, for VoIP applications, a packet loss rate below 1% is still toll quality [152]. If the loss rate is over 5%, then the voice quality is bad [153] [77]. A loss rate beyond 20% is beyond usefulness for VoIP [99]. So, CLR is also used as a preliminary criterion for judging the path switching control results for the VoIP flow. Fig. 37 (b) gives an example of the CLR signal for one of the trace-files of Path A. It is clear that the CLR signal reflects both the losses and the over-delays.

b. Forward delays

Another type of information signal used for comparing path quality is the *forward delay*. The forward delay of a path at time t is the amount of time a packet needs to be transmitted over that path and to reach the destination if it is sent at that time. In this study, the delay measurement of a packet is associated with its send-time t_{send}. The delay measurement of a packet is calculated as the difference between its send-time t_{send} at the send side and its receive-time t_{recv} at the receive side:

$$d_f(t_{send}, SN) = t_{recv}(SN) - t_{send}(SN), \tag{7.19}$$

where SN is the sequence number of the packet. The measured delays are averaged every T_p and are predicted two-step-ahead for the predictive path switching control.

Two issues need to be taken care of when calculate the average delay for every prediction interval: how to account for the lost packets and how to deal with the over-delayed packets. For a lost packet, there is no delay associated with it. Its delay value is assigned to be the information feedback delay limit ,$\tau = 400$ ms, plus a safety margin. In this study this delay value is set to be 550 ms.

For an over-delayed packet, if its delay is beyond the one-way delay limit $\tau_f = 150$ ms, but below the information feedback delay limit $\tau = 400$ ms, then although it is over-delayed its probing information can still be fed back to the controller. so its delay value is kept the same. If the delay of a packet is more than $\tau = 400\ ms$, then this delay information can not be feedback to the predictive path switching controller in time. There is no way to distinguish this packet from a lost packet in real time. So the delay associated with this packet is also assigned with a delay value of 550 ms. Fig. 37 (c) gives an example of the delay signal of the same trace-file of Path A. It demonstrates the effect of assigning a delay value to the lost packets.

c. Accumulation

The third type of information signal used to indicate path quality is the *accumulation*. The accumulation of a flow in a path at a given time t is defined as the amount of bytes of that flow in transit inside that path at that time. The difference between the cumulative send flow $U(t)$ and the cumulative arrival flow $Z(t)$ is called the accumulation-and-losses signal $AL(t)$. It is given by

$$AL(t) = U(t) - Z(t). \tag{7.20}$$

This signal contains both the accumulation signal $Acc(t)$, and the cumulative losses signal $L(t)$:

$$AL(t) = Acc(t) + L(t). \tag{7.21}$$

In this study these signals are given in terms of number of bytes.

In order to separate the accumulation signal $Acc(t)$ and the cumulative losses signal $L(t)$, the accumulation-and-losses signal $AL(t)$ needs to be detrended. The increasing trend in the $AL(t)$ signal is mostly because of the cumulative losses signal

$L(t)$. Khariwal detrended the $AL(t)$ signal in batch mode by removing the linear increasing trend from it [130]. Shukla detrended it by obtaining the trend through a low pass filter and removing it from the signal [154]. An alternative way to detrend this signal is to obtain the cumulative losses signal $L(t)$ directly by finding out the occurrence of a lost packet as soon as possible, and remove the losses signal $L(t)$ from the accumulation-and-losses signal $AL(t)$. This can be done by setting a time limit for a packet arrive. If this packet has not arrived in time, then it is considered lost and removed from the $AL(t)$ signal. When there is no flow reversal, i.e. no packet reordering, receiving a packet with sequence number $SN = k$ means all the packets with sequence numbers $SN < k$ and have not arrived are lost. Kommaraju used this method to remove the trend from the $AL(t)$ signal [123]. In this research, Kommaraju's detrending method is used.

The accumulation signal is different from the CLR and delay signals. It is not only related to the property of a particular path, but also related to the send rate of the flow. In this research, the accumulation signals of the probing flows in the path, are used as the information signals for predictive path switching control. The probing flows have constant send rates, so the variations in the accumulation signals reflect only the changes in the path's condition.

Theoretically the accumulation signal is available at all time and can be sampled at any rate. The sampling rate of CLR and delay signals are restricted to the probing rate. Also the measurement of the delay signal is not available when a probing packet is lost.

Xia et al. [17] used accumulation for congestion control. They have proved that TCP Vegas is in fact equivalent to a control based on the accumulation signal. Khariwal [130] used the accumulation signal for adaptive control of the send rate of multimedia applications in best-effort networks. In this research the accumulation signal

is sampled every 100 ms. The accumulation signal is averaged every 400 ms to be used in predictive path switching control.

Fig. 37 (d) gives an example of the accumulation signal of a trace-file of Path A. Fig. 38 is a zoom-in comparison of the delays of the trace-file and the corresponding accumulation signal. Note that the accumulation signal has very similar shape compared to the delays in the trace-file. But when the delay is high, the corresponding peak in the accumulation signal is lagging behind.

d. Auto-correlation of the information signals

The autocorrelation of these information signals will provide an idea of the linear dependency of these signals on their past. It is very help in determine the order of linear predictors for these signals. Given the measurements $y_{i,T_p}(1), y_{i,T_p}(2), \ldots, y_{i,T_p}(N)$ (i=A,B,C) of a path, at time $t(1), t(2), \ldots, t(N)$, the autocorrelation function [156] is defined as:

$$r(k) = \frac{\sum_{j=1}^{N-k} \left(y_{i,T_p}(j) - \bar{y}_{i,T_p}\right)\left(y_{i,T_p}(j+k) - \bar{y}_{i,T_p}\right)}{\sum_{j=1}^{N} \left(y_{i,T_p}(j) - \bar{y}_{i,T_p}\right)^2}, \tag{7.22}$$

where, k is the number of lags, and

$$\bar{y}_{i,T_p} = \frac{1}{N} \sum_{j=1}^{N} y_{i,T_p}(j). \tag{7.23}$$

The autocorrelation plots of the information signals in Fig. 37 are given in Fig. 39. It can be seen that the autocorrelations drop to 0.2 after 300 lags in all three signals, which indicates a long term dependency in the signals. Fig. 40 shows the autocorrelation plots of the information signals of another trace-file. This time the autocorrelation function of CLR drops below 0.2 after only 10 lags, that of delay drops below 0.2 after 20 lags, while that of accumulation drops below 0.2 after 30 lags. These results indicate that there is a lot of diversity in the linear dependency of these signals

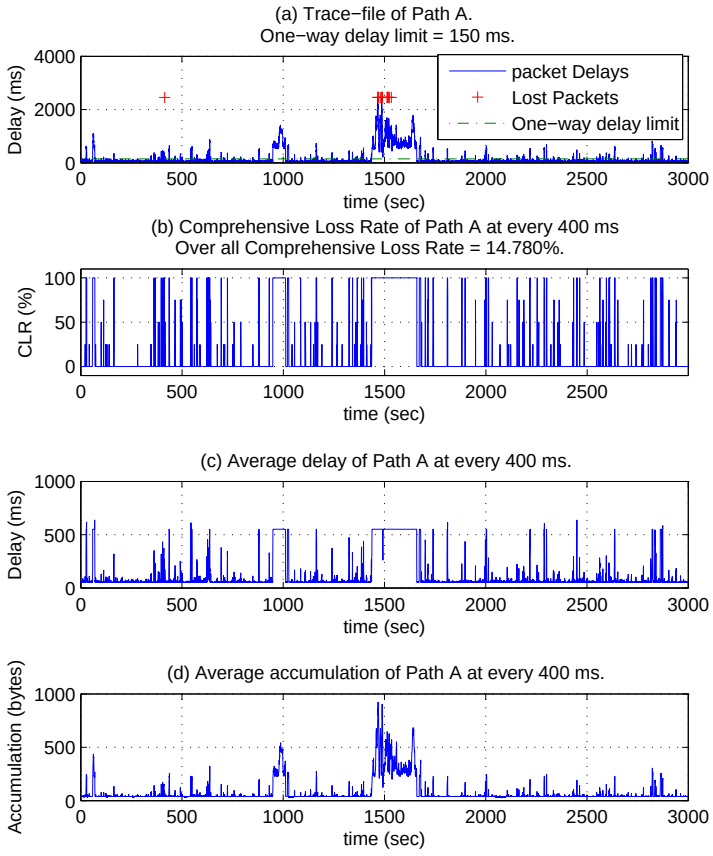

Fig. 37. An example of the trace-file and its corresponding information signals. (a) One trace-file of Path A. (b) The CLR signal of this trace-file used in predictive path switching. (c) The delay signal of this trace-file used in predictive path switching. (C) The accumulation signal of this trace-file used in predictive path switching.

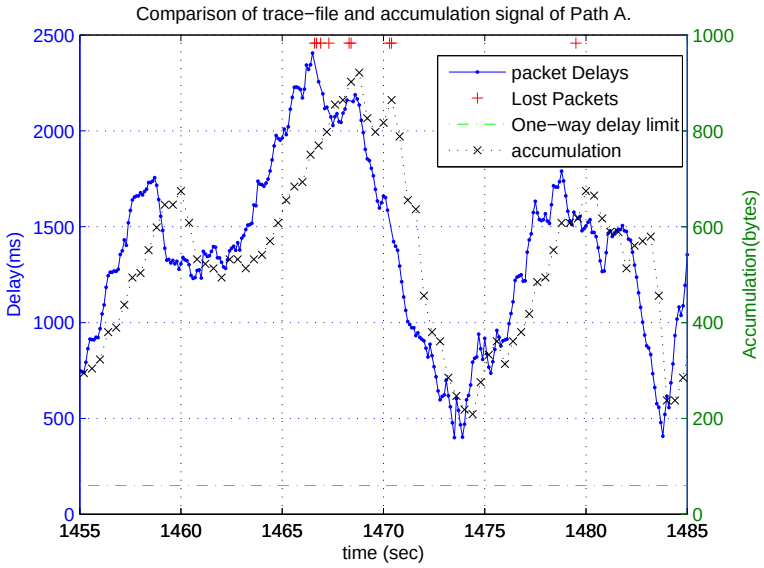

Fig. 38. A zoom-in comparison of the trace-file and its corresponding accumulation signal.

on their past. It is very difficult to use one predictor for the predictions of all the information signals from difference trace-files.

2. Different types of predictors

According to the assumption, the controlled VoIP flow and the probing flows are all very small compared to the cross flows in the network. The information signals are mainly dependent on the cross traffic. The controlled VoIP flow and the probing flows do not effect the information signals. The information signals are viewed as time series in this research. Because of lack of information of the the cross flows and lack of access to the routers inside the networks, a white box model is very difficult to obtain. Only black box models are developed for the prediction of these information signals. The structures of the five types of predictors used in this research are presented as follows.

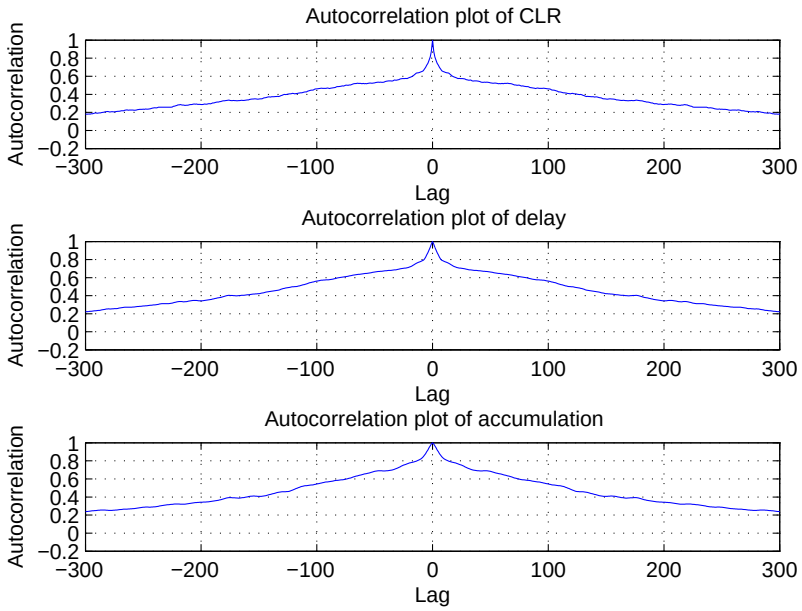

Fig. 39. Autocorrelation plot of the information signals of one trace-file. Top: Autocorrelation plot of the CLR signal. Middle: Autocorrelation plot of the delay signal. Bottom: Autocorrelation plot of the accumulation signal.

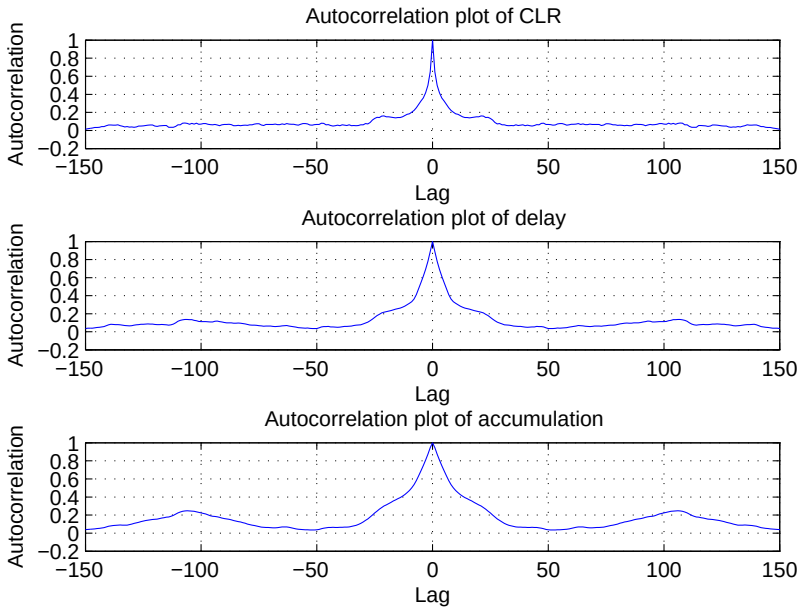

Fig. 40. Autocorrelation plot of the information signals of another trace-file. Top: Autocorrelation plot of the CLR signal. Middle: Autocorrelation plot of the delay signal. Bottom: Autocorrelation plot of the accumulation signal.

All five type of predictors are developed for each of the three types of information signal.

a. Simple predictor

The first type of predictor is simple predictor (SP). SP uses the most resent measurements as the predictions of the future values. For prediction step size $T_p = 400$ ms, the two-step-ahead simple predictor is given by

$$\hat{y}_{i,400,SP}(k|k-2) = y_{i,400}(k-2). \tag{7.24}$$

A controller based on SP is in fact an reactive controller. The predictions of this predictor and the control results based on this predictor are used as base lines to compare to other predictors.

b. AutoRegressive predictor

The AutoRegressive (AR) model of a time series is a linear model of a random process which has the form [157]:

$$(1 - a_1 q^{-1} - a_2 q^{-2} - \cdots - a_{n_a} q^{-n_a})\tilde{y}(k) \;=\; A(q^{-1})\tilde{y}(k) = v(k), \tag{7.25}$$

where, n_a is the model order, q^{-1} is the backward shift operator, i.e. $q^{-j}\tilde{y}(k) = \tilde{y}(k-j)$,

$$\left\{ \tilde{y}(k) = y(k) - \frac{1}{N}\sum_{j=1}^{N} y(j) \right\} \tag{7.26}$$

is the deviation of the time series from its mean, $\{v(k)\}$ is some white noise input, and $A(q^{-1})$ is the transfer function of the linear filter relating $\tilde{y}(k)$ to $v(k)$. The AR model can also be written as

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2) + \cdots + a_{n_a} y(k-n_a) + v(k), \tag{7.27}$$

where $\{y(k)\}$ is the time series, $v(k)$ is white noise, and $a_0 = (1 - \sum_{j=1}^{n_a} a_j)\mu$ with

$$\mu = \frac{1}{N}\sum_{j=1}^{N} y(j) \tag{7.28}$$

denoting the mean of the time series [156].

Based on the AR model of a time series, one-step-ahead predictor of that time series can be written as [158]

$$\hat{y}(k|k-1) = a_0 + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a). \tag{7.29}$$

Starting from the one-step-ahead predictor, a d-step-ahead predictor can be obtained through iteration.

$$\hat{y}(k|k-d) = a_0^{\{d\}} + a_1^{\{d\}} y(k-d) + \cdots + a_{n_{a_d}}^{\{d\}} y(k-d-n_{a_d}+1), \tag{7.30}$$

where n_{a_d} is the order of the predictor, and $\{a_j^{\{d\}}, j = 0, 1, \ldots, n_{a_d}\}$ are the coefficients. Equation (7.30) can be put into regression form

$$\hat{y}(k|k-d) = \phi^T(k-d)\theta, \tag{7.31}$$

$$\phi(k-d) = \begin{bmatrix} 1 & y(k-d) & y(k-d-1) & \cdots & y(k-d-n_{a_d}+1) \end{bmatrix}^T, \tag{7.32}$$

$$\theta = \begin{bmatrix} a_0^{\{d\}} & a_1^{\{d\}} & a_2^{\{d\}} & \cdots & a_{n_{a_d}}^{\{d\}} \end{bmatrix}^T \tag{7.33}$$

where, ϕ is a vector of known quantities called regressors, and θ is a vector of unknown parameters. Instead of using the iteration method, the parameters $\{a_j^{\{d\}}, j = 0, 1, \ldots, n_{a_d}\}$ can also be obtained through the least squares estimation. This is done by directly minimizing the quadratic d steps ahead prediction errors

$$\epsilon(k) = y(k) - \hat{y}(k|k-d), \tag{7.34}$$

$$V(\theta) = \frac{1}{2}\sum_{k=1}^{N} \epsilon^2(k). \tag{7.35}$$

In this research, the two-step-ahead AR predictor is given by

$$\begin{aligned}\hat{y}_{i,400,AR}(k|k-2) &= a_{0,i,AR} + a_{1,i,AR}y_{i,400}(k-2) \\ &\quad + \cdots + a_{n_{a,i},i,AR}y_{i,400}(k-n_{a,i}-1). \end{aligned} \tag{7.36}$$

The parameters are estimated using least squares method

$$\hat{\theta}_{i,AR} = (\mathbf{\Phi}_i^T\mathbf{\Phi}_i)^{-1}\mathbf{\Phi}_i\mathbf{y}_i, \tag{7.37}$$

where

$$\theta_{i,AR} = \begin{bmatrix} a_{0,i,AR} & a_{1,i,AR} & \cdots & a_{n_{a,i},i,AR} \end{bmatrix}^T, \tag{7.38}$$

$$\mathbf{\Phi}_i = \begin{bmatrix} 1 & y_i(n_{a,i}) & \cdots & y_i(1) \\ 1 & y_i(n_{a,i}+1) & \cdots & y_i(2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_i(N-2) & \cdots & y_i(N-n_{a,i}-1) \end{bmatrix}, \tag{7.39}$$

$$\mathbf{y}_i = \begin{bmatrix} y_{i,400}(n_{a,i}+2) & y_{i,400}(n_{a,i}+3)\cdots & y_{i,400}(N) \end{bmatrix}^T. \tag{7.40}$$

The AR predictors are linear predictors which can be quickly developed to reveal some characteristics of the information signals. The AR predictors are also easy to implement. In this research, the parameter to be tuned for a AR predictor is the maximum order of the predictor (n_a).

c. Nonlinear AutoRegressive predictor

It is observed that there is statistical self-similarity in both local-area network (LAN) traffic [159] and wide-area network (WAN) traffic [160]. Borella et al. [9] reported self-similarity in the Internet packet delay. These reports give a strong indication of nonlinearity. Hasegawa et al. [120] showed that it is worth trying nonlinear time

series prediction methods on the Internet traffic data. Nonlinear predictors are also used in this study.

Most nonlinear dynamic systems can be described using the Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) model

$$\begin{aligned} y(k) &= f\left(y(k-1),\ldots,y(t-n_y),u(t-1),\ldots,u(t-n_u),\right. \\ &\quad \left. e(k-1),\ldots,e(k-n_e)\right)+e(k), \end{aligned} \tag{7.41}$$

where $y(k)$, $u(k)$, $e(k)$ are the system output, input, and noise sequences, respectively, $f(\cdot)$ is some non-linear function, n_y, n_u, and n_e are the maximum lags in the output, input and noise, and $e(k)$ is white noise sequence. Approximating $f(\cdot)$ near the operating point with high order polynomial functions gives a polynomial NARMAX model representation of the non-linear system [161].

In this study, a special case of the general NARMAX model equation (7.41), i.e. Nonlinear AutoRegressive (NAR) model, is used to model the time series:

$$y(k) = f\left(y(k-1),\ldots,y(k-n_y)\right)+e(k). \tag{7.42}$$

Expanding $f(\cdot)$ as a polynomial of degree l gives the representation [162]

$$\begin{aligned} y(k) &= \theta_0+\sum_{i_1=1}^{n_y}\theta_{i_1}y(k-i_1)+\sum_{i_1=1}^{n_y}\sum_{i_2=i_1}^{n_y}\theta_{i_1i_2}y(k-i_1)y(k-i_2)+\ldots \\ &\quad +\sum_{i_1=1}^{n_y}\cdots\sum_{i_l=i_{l-1}}^{n_y}\theta_{i_1\ldots i_l}y(k-i_1)\ldots y(k-i_l)+e(k). \end{aligned} \tag{7.43}$$

A two step ahead polynomial NAR predictor of degree 2 becomes

$$\hat{y}(k|k-2) = \theta_0+\sum_{i_1=2}^{n_y}\theta_{i_1}y(k-i_1)+\sum_{i_1=2}^{n_y}\sum_{i_2=i_1}^{n_y}\theta_{i_1i_2}y(k-i_1)y(k-i_2). \tag{7.44}$$

The system can be put into a linear regression model [162]

$$y(k) \;=\; \hat{y}(k|k-2) + \xi(k) = \sum_{j=1}^{M} p_j(k)\theta_j + \xi(k), \quad k = n_y + 1, \ldots, N, \quad (7.45)$$

where N is the data length, the $p_j(k)$ are monomial of $y(k-2)$ to $y(k-n_y)$ up to degree 2, $p_1(k) = 1$ corresponding to a constant term, M is the number of $p_j(k)$ terms, $\xi(k)$ is the prediction error, and the θ_j are unknown parameters to be estimated. Equation (7.45) can be written in the matrix form [162]

$$\mathbf{y} = \mathbf{P}\boldsymbol{\Theta} + \boldsymbol{\Xi}, \quad (7.46)$$

with

$$\begin{aligned}
\mathbf{y} &= \begin{bmatrix} y(n_y+1) & \ldots & y(N) \end{bmatrix}^T, \\
\mathbf{P} &= \begin{bmatrix} \mathbf{p}_1 & \ldots & \mathbf{p}_M \end{bmatrix}, \\
\mathbf{p}_j &= \begin{bmatrix} p_j(n_y+1) & \ldots & p_j(N) \end{bmatrix}^T, j = 1, \ldots, M, \\
\boldsymbol{\Theta} &= \begin{bmatrix} \theta_1 & \ldots & \theta_M \end{bmatrix}^T, \\
\boldsymbol{\Xi} &= \begin{bmatrix} \xi(n_y+1) & \ldots & \xi(N) \end{bmatrix}^T. \quad (7.47)
\end{aligned}$$

A least squares (LS) problem is formed to estimate the parameters $\boldsymbol{\Theta}$ which minimizes

$$V(\boldsymbol{\Theta}) = \| \mathbf{y} - \mathbf{P}\boldsymbol{\Theta} \| = \frac{1}{2}(\mathbf{y} - \mathbf{P}\boldsymbol{\Theta})^T(\mathbf{y} - \mathbf{P}\boldsymbol{\Theta}). \quad (7.48)$$

The solution to the optimization problem satisfies the normal equation [162]

$$\mathbf{P}^T\mathbf{P}\boldsymbol{\Theta} = \mathbf{P}^T\mathbf{y}, \quad (7.49)$$

where $\mathbf{P}^T\mathbf{P}$ is called the information matrix. In reality, only a few significant terms will characterize the system dynamics. The combined problem of structure selecting

and parameter estimation (SSPE) is shown in the paper by Chen et al. [162]. The problem statement can be written as:

> Select a subset $\mathbf{P}_s$ of $\mathbf{P}$ and find the corresponding parameter estimate $\hat{\Theta}_s$

As shown by Chen et al. the solution to this problem can be found by using the Modified Gram-Schmidt (MGS) method.

Assume that $\mathbf{P_s}$ has M_s columns. Try to factorize it as

$$\mathbf{P}_s = \mathbf{W}_s \mathbf{A}_s, \tag{7.50}$$

where

$$\begin{aligned} \mathbf{W}_s &= \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_{M_s} \end{bmatrix} \\ &= \begin{bmatrix} w_1(1) & \cdots & w_{M_s}(1) \\ \vdots & & \vdots \\ w_1(N-n_y) & \cdots & w_{M_s}(N-n_y) \end{bmatrix} \end{aligned} \tag{7.51}$$

is an $(N - n_y) \times M_s$ orthogonal matrix of M_s columns and

$$\mathbf{A}_s = \begin{bmatrix} 1 & \alpha_{1\,2} & \alpha_{1\,3} & \cdots & \alpha_{1\,M_s} \\ & 1 & \alpha_{2\,3} & \cdots & \alpha_{2\,M_s} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \alpha_{M_s-1,M_s} \\ & & & & 1 \end{bmatrix} \tag{7.52}$$

is an $M_s \times M_s$ unit upper triangular matrix. Then

$$\mathbf{W}_s^T \mathbf{W}_s = \mathbf{D}_s, \tag{7.53}$$

where $\mathbf{D}_s$ is a positive diagonal matrix. Define

$$\mathbf{g}_s = \mathbf{D}_s^{-1}\mathbf{W}_s^T\mathbf{y}, \tag{7.54}$$

the parameter estimate $\hat{\Theta}_s$ can be computed from

$$\mathbf{A}_s\mathbf{\Theta}_s = \mathbf{g}_s. \tag{7.55}$$

The residuals are given by

$$\begin{aligned} \hat{\Xi} &= \mathbf{y} - \mathbf{P}_s\hat{\mathbf{\Theta}}_s \\ &= \mathbf{y} - \mathbf{W}_s\mathbf{g}_s. \end{aligned} \tag{7.56}$$

The error reduction ratio due to $\mathbf{w}_i$ is defined as

$$\epsilon_i = \frac{g_i^2 < \mathbf{w}_i, \mathbf{w}_i >}{< \mathbf{y}, \mathbf{y} >}, \tag{7.57}$$

where $< \cdot, \cdot >$ denotes the inner product. The MGS method is given as follows:

1. Set $I_1 = \{1, 2, \ldots, M\}$; // Create the initial set of candidate terms.

   $j = 1$;

   for $i = 1$ to M

   $\mathbf{p}_i^{(0)} = \mathbf{p}_i$;

   $\mathbf{w}_1^{(i)} = \mathbf{p}_i^{(0)}$;

   $g_1^{(i)} = \dfrac{< \mathbf{w}_1^{(i)}, \mathbf{y} >}{< \mathbf{w}_1^{(i)}, \mathbf{w}_1^{(i)} >}$;

   $\epsilon_1^{(i)} = \dfrac{(g_1^{(i)})^2 < \mathbf{w}_1^{(i)}, \mathbf{w}_1^{(i)} >}{< \mathbf{y}, \mathbf{y} >}$;

   end for

   $l_1 = \arg\max_{i \in I_1}\{\epsilon_1^{(i)}\}$; // Find the first term which gives the maximum error reduction.

   $\mathbf{w}_1 = \mathbf{w}_1^{(l_1)}$;

$g_1 = g_1^{(l_1)}$;

$\epsilon_1 = \epsilon_1^{(l_1)}$;

$\alpha_{1\,1} = 1$;

2. $j = j + 1$;

$I_j = I_{j-1} \backslash \{l_{j-1}\}$; //Remove the picked term from the set of candidate terms.

for all $i \in I_j$

$$\tilde{\alpha}_{j-1\,i} = \frac{< \mathbf{w}_{j-1}, \mathbf{p}_i^{(j-2)} >}{< \mathbf{w}_{j-1}, \mathbf{w}_{j-1} >};$$

$$\mathbf{p}_i^{(j-1)} = \mathbf{p}_i^{(j-2)} - \tilde{\alpha}_{j-1\,i} \mathbf{w}_{j-1};$$

$$\mathbf{w}_j^{(i)} = \mathbf{p}_i^{(j-1)};$$

$$g_j^{(i)} = \frac{< \mathbf{w}_j^{(i)}, \mathbf{y} >}{< \mathbf{w}_j^{(i)}, \mathbf{w}_j^{(i)} >};$$

$$\epsilon_j^{(i)} = \frac{(g_j^{(i)})^2 < \mathbf{w}_j^{(i)}, \mathbf{w}_j^{(i)} >}{< \mathbf{y}, \mathbf{y} >};$$

end for

$J_j = \left\{ \arg_{i \in I_j} (\mathbf{w}_j^{(i)^T} \mathbf{w}_j^{(i)} < Th^1) \right\}$;

$I_j = I_j \backslash J_j$; // Remove the terms which are too small from the set of candidate terms

$l_j = \arg \max_{i \in I_j} \{\epsilon_j^{(i)}\}$; // Find the next term which gives the maximum error reduction.

$\mathbf{w}_j = \mathbf{w}_j^{(l_j)}$;

$g_j = g_j^{(l_j)}$;

$\epsilon_j = \epsilon_j^{(l_j)}$;

$\alpha_{j\,j} = 1$;

for $k = 1$ to $j - 1$

$\alpha_{k\,j} = \tilde{\alpha}_{k\,l_j}$;

end for

3. Repeat step 2 until the Akaike's information criterion

$$AIC(4) = (N - n_y) \log \left(\frac{1}{N - n_y} < \hat{\Xi}, \hat{\Xi} > \right) + M_s \times 4 \tag{7.58}$$

reaches a minimum.

The polynomial NAR predictor is linear in parameters. Although the selection of significant terms and determination of coefficients are complicated, the implementation complexity of the NAR predictor is almost the same as that of the AR predictor. In this research, polynomials of degree up to 2 is used. The parameters to be tuned are the maximum lags used (n_y), and the number of terms used (M_s).

d. Radial basis function predictor

Neural networks are widely used for modeling nonlinear systems. Theoretically, given sufficient number of hidden neurons, neural networks can approximate any continuous function with arbitrary accuracy. But normally neural networks are highly nonlinear in parameters. The radial basis function (RBF) network is an alternative to highly non-linear-in-parameter neural networks [163]. It is linear in parameters. The nonlinearities are modeled by the fixed nonlinear transformation in the hidden layer. Kommaraju [123] reported that the RBF based predictors give better predictions of the network accumulation in terms of number-of-packets compared to SP and AR based predictors.

In this research, the RBF tools from the neural networks toolbox of Matlab [164] is used to train the network. In Matlab, the structure of the RBF network is given as in Fig. 41. The vector distance between the input vector **p** and the center vector

[1] $Th = 10^{-10}$ is a threshold to prevent numerical ill conditioning.

$\mathbf{w}_i$ of each neuron i is calculated and multiplied by a bias b_1.

$$
\begin{aligned}
n_i &= \| \mathbf{w}_i - \mathbf{p} \| b_1, && (7.59)\\
b_1 &= \frac{0.8326}{\beta}, && (7.60)\\
\mathbf{p} &= \begin{bmatrix} y(k-2) & y(k-3) & \dots & y(k-n_y) \end{bmatrix}^T, && (7.61)
\end{aligned}
$$

where, β is the spread of the radial basis function, and n_y is the maximum lags used in the input vector. The result n_i is passed to the radial basis neuron transfer function of each neuron to generate the output $a_i^{(1)}$:

$$a_i^{(1)} = e^{-(n_i)^2}. \tag{7.62}$$

Finally the linear combination of the outputs of these neurons plus the output bias b_2 gives the two-step-ahead prediction $\hat{y}(k|k-2)$.

$$\hat{y}(k|k-2) = \theta^T \mathbf{a}^{(1)} + b_2. \tag{7.63}$$

In Fig. 41,

$$
\begin{aligned}
\mathbf{IW}_{1,1} &= \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_{S^{(1)}}^T \end{bmatrix}, && (7.64)\\
\mathbf{LW}_{2,1} &= \theta^T. && (7.65)
\end{aligned}
$$

In order to make the tuning of parameters easier, the information signals are scaled down by a factor α, before the RBF network models are developed for the signals. This factor α is equal to the mean, μ, of the signals plus three times of the standard

deviation σ of the signals,

$$\begin{aligned} \mu &= E\{y\}, \\ \sigma &= \sqrt{E\{(y-\mu)^2\}}, \\ \alpha &= \mu + 3\sigma, \end{aligned} \tag{7.66}$$

where $E\{\cdot\}$ is the expectation. When using this predictor the signal feeding into the predictor is scaled down by the factor α. The output of the prediction is scaled back by the same factor α to compare with the results of other predictors. The parameters tuned are the maximum lags used (n_y), the number of neurons in the hidden layer ($S^{(1)}$), and the spread of the radial basis function (β).

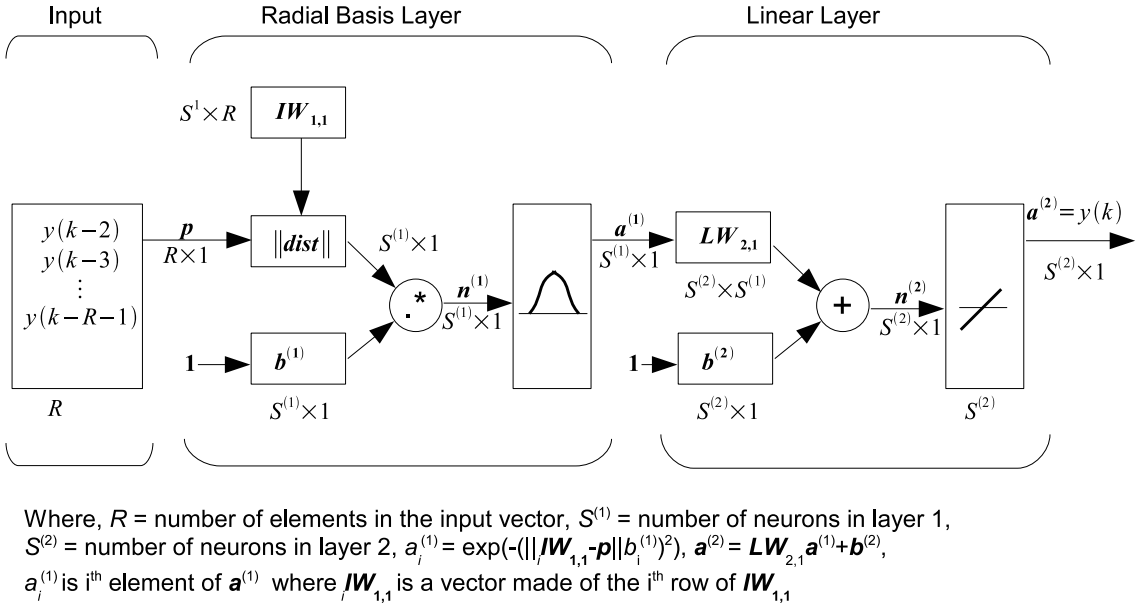

Fig. 41. Structure of the RBF network in Matlab.

e. Ad hoc predictor

During the development and testing of predictors, it is realized that the predictors which give the best predictions in the MSE sense, does not always perform the best for predictive path switching control. It is also observed that a prediction combining the most recent available information and an average of some past information should also be a good indicator of path quality for predictive path switching control. The following ad hoc (Adhoc) predictor structure is thus proposed:

$$\hat{y}(k|k-2) = ay(k-2) + (1-a)\frac{1}{N}\sum_{i=0}^{N-1} y(k-2-i). \tag{7.67}$$

In this study, $N = 80$ is attempted. Different values of a are directly used to generate predictions for the predictive path switching control. The a value which gives the best predictive path switching control results is picked.

3. Parameter selection

In this research, a data set for predictor development is split into three parts. The first part is the training set. Given one set of model parameters, this training set is used to determine the coefficients for a selected model structure. The second part is the testing set. It is used to compare the prediction results of different model parameters. The third part is the validation set. It is used to compare the different model structures.

In this research, all the five types of predictors are built for each of the three types of information signals. Four predictors of the same model structure for the same type information signal are developed for each path. Each of the four predictors is developed from one of the first four trace-files of a path. Each of the first four trace-files of a path is divided into three parts. The first part is used as the training set of

the predictor, the second part as the testing set, and the third part as the validation sets. The MSE defined in equation (7.17) is used as the criterion for validation.

For the SPs there are no parameters to be tuned. For the AR, NAR, and RBF predictors, their final parameters are given in Table XVI. The tuned parameters are maximum order n_a for AR predictors; maximum lags $n_{y,NAR}$ and number of terms M_s for NAR predictors; maximum lags n_y, number of hidden layers $S^{(1)}$, and spread β for RBF predictors. The scaling factors α for RBF predictors are also given in this table. The ad hoc predictors are tuned directly based on their predictive path switching control results, so their weighting factors depend on whether the switching is between path pair pair AB or path pair AC. These weighting factors are also listed in Table XVI.

For predictors using AR, NAR, and RBF model structures, the four predictors of the same model structure, which are developed for the same path and same type of information signal, are cross validated on each other's validation set. The cross validation results (MSE) are given in Table XVII.

Comparing the parameters of the four predictors of the same structure for the same path and same type of information signal in Table XVI, it is clear that the parameters obtained from different trace-files are very different. Comparing the cross validation results of the four predictors of the same model structure for the same path and same type of information signal in Table XVII, in most cases, the results of the four predictors of give resulting MSEs in about the same range, but no one predictor is the best. It is very difficult to say that which set of parameters is the best.

4. Prediction results

The average prediction results of the four predictors of same model structure for the same path and same type of information signal are used, to compare the prediction

Table XVI. Predictor parameters.

Signal	Path	Trace-file	SP	AR	NAR		RBF				Adhoc
			N/A	n_a	$n_{y,NAR}$	M_s	α	$n_{y,RBF}$	$S^{(1)}$	β	a
CLR	A	1		39	28	4	1.28	30	65	16	
		2		2	1	1	0.75	10	40	8	
		3		4	28	3	0.85	10	65	2	
		4		33	21	4	0.89	10	40	1	
	B	1		43	46	11	1.46	10	65	16	0.71[1]
		2		24	16	6	1.16	40	30	16	0.70[1]
		3		28	12	6	1.28	10	50	1	0.71[1]
		4		36	27	3	1.48	10	40	8	0.80[1]
	C	1		18	18	4	1.28	10	30	1	0.73[2]
		2		1	25	3	0.64	30	40	16	0.70[2]
		3		7	14	3	0.85	10	40	4	0.70[2]
		4		32	1	1	0.80	10	80	8	0.60[2]
Delay	A	1		52	28	5	583.42	50	30	8	
		2		41	1	2	354.74	10	65	4	
		3		3	1	1	380.56	10	50	1	
		4		17	43	6	400.52	10	30	2	
	B	1		37	50	9	615.13	10	50	8	0.66[1]
		2		33	16	9	461.35	10	50	8	0.40[1]
		3		19	50	11	536.84	10	40	16	0.65[1]
		4		52	28	5	599.52	10	30	16	0.56[1]
	C	1		19	15	9	639.44	40	30	8	0.67[2]
		2		93	2	2	289.25	20	65	1	1.00[2]
		3		8	15	6	348.98	10	50	1	0.69[2]
		4		32	4	4	363.43	10	40	2	0.73[2]
Accum.[3]	A	1		52	32	7	378.79	30	30	16	
		2		35	41	8	171.17	20	30	8	
		3		2	7	3	177.16	50	30	16	
		4		1	30	7	222.15	10	30	16	
	B	1		5	1	1	465.05	30	50	16	0.99[1]
		2		32	37	8	222.50	10	50	16	0.85[1]
		3		12	50	5	290.42	10	50	1	0.95[1]
		4		21	32	7	329.74	40	30	16	0.95[1]
	C	1		19	43	29	433.35	20	65	2	1.00[2]
		2		84	2	2	130.91	20	40	1	1.00[2]
		3		1	17	6	205.40	10	30	2	1.00[2]
		4		34	38	4	162.93	10	40	4	0.97[2]

[1]For switching between paths A and B.

[2]For switching between paths A and C.

[3]Accumulation signal.

Table XVII. Cross validations of predictors of the same varieties.

Trace-file	Cross Validation MSE(%)[1]											
	AR Predictors				NAR Predictors				RBF Predictors			
CLR	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4
A1	60.10[2]	60.64	59.62	58.90	48.33	51.81	55.14	50.69	59.04	58.26	59.75	60.00
A2	62.38	60.77	60.48	62.90	57.80	57.71	57.04	56.54	62.13	57.74	60.88	59.96
A3	84.92	86.89	81.56	83.28	36.04	35.93	37.09	36.85	85.15	78.36	83.47	80.09
A4	59.00	57.77	57.00	57.95	65.19	67.63	66.55	64.48	58.35	54.75	55.31	57.44
	B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4
B1	74.95	70.80	72.53	71.90	72.74	67.70	68.11	73.66	50.10	51.15	52.71	50.50
B2	69.03	66.80	68.22	67.95	67.30	63.71	64.84	70.84	63.11	60.73	63.37	65.13
B3	67.45	65.52	65.21	66.34	64.59	62.85	62.67	68.68	44.22	43.86	42.62	46.12
B4	57.35	57.72	58.10	57.42	56.82	56.92	58.26	55.90	45.40	47.27	49.87	45.65
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
C1	65.45	61.10	61.29	62.33	59.99	68.90	61.45	61.23	63.77	65.94	67.00	63.92
C2	79.92	72.90	73.29	74.81	72.66	72.89	72.62	73.95	69.15	70.75	70.59	71.77
C3	73.39	67.20	66.81	68.37	67.60	67.55	67.97	67.81	61.28	61.55	58.28	61.19
C4	77.35	68.41	70.04	72.93	69.12	67.99	70.22	69.11	53.82	56.59	57.12	57.35
Delay	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4
A1	27.60	27.86	28.05	27.58	25.49	25.98	27.51	28.50	26.99	34.14	28.74	28.21
A2	29.32	28.58	29.31	29.07	23.46	22.40	23.91	22.45	28.65	26.61	27.69	28.52
A3	35.22	35.73	35.88	37.25	18.47	16.71	17.60	16.48	33.66	30.98	37.03	34.14
A4	26.93	26.44	26.53	26.84	26.74	25.72	27.32	26.70	26.00	26.22	27.28	24.84
	B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4
B1	37.67	36.09	37.71	35.34	36.35	34.06	35.76	36.70	25.79	25.34	27.39	26.08
B2	32.01	30.97	31.90	31.86	31.39	29.11	29.90	33.24	33.07	27.80	30.05	32.65
B3	29.07	28.96	29.10	27.97	28.01	26.59	27.82	29.92	16.86	15.58	15.40	18.43
B4	29.98	29.89	30.53	30.23	29.08	28.85	29.52	29.12	26.06	26.94	27.95	25.37
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
C1	32.30	31.01	31.37	31.69	31.53	31.05	31.28	31.29	33.23	45.20	33.65	30.53
C2	32.72	31.59	32.47	32.06	29.23	31.35	30.95	32.43	32.30	29.43	28.94	29.23
C3	33.82	32.49	32.76	32.81	31.60	32.78	32.27	32.17	30.18	34.51	28.16	29.34
C4	33.30	31.97	32.96	33.31	31.06	30.54	32.04	33.20	30.74	44.01	28.51	28.45
Accum.	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4
A1	18.71	19.04	19.10	19.51	17.39	22.32	18.08	16.45	18.42	19.61	17.52	20.13
A2	18.67	18.52	18.88	19.30	14.33	13.68	14.26	13.05	18.83	18.30	17.59	19.82
A3	21.08	20.60	20.73	20.86	12.30	11.90	12.24	11.25	19.85	19.96	18.41	19.49
A4	17.37	17.44	17.69	18.02	14.95	21.69	15.37	13.26	17.89	17.37	16.27	18.32
	B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4
B1	23.69	21.10	22.70	21.25	22.99	18.78	20.57	20.64	10.88	533.69	53.50	15.88
B2	19.92	18.27	19.12	18.44	20.05	16.74	17.85	18.18	19.88	15.69	17.21	18.64
B3	16.56	16.08	16.35	16.01	16.90	15.49	15.25	15.74	11.14	10.09	9.64	11.80
B4	21.07	19.62	20.36	19.66	20.89	18.01	18.80	18.36	14.21	72.42	30.37	13.21
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
C1	16.56	16.38	17.04	16.39	16.94	23.97	14.38	21.04	19.66	27.77	20.89	20.12
C2	16.49	14.90	16.58	15.20	15.51	14.07	14.14	14.78	15.30	13.48	13.45	14.76
C3	16.87	15.48	16.60	15.49	15.97	15.39	14.70	15.36	18.50	33.75	16.60	24.83
C4	16.17	14.34	16.07	14.89	14.99	13.75	13.82	14.00	16.53	28.85	18.80	14.83

[1] The minimum MSE of the four predictors of each variety on each trace-file is framed.

[2] This cell means it is the MSE of the prediction of CLR signal of the AR predictor, which is developed from the training set of the first trace-file of Path A and is test on the validation set of the first trace-file of Path A. The meaning of the other cells are explained in the same way.

results of different types of predictors. The predictors are developed using the data from trace-files of paths A, B, and C. To compare the consistency of the predictors, three new sets of data are collected. Each data set consists of 9 one hour long trace-files. These data sets are collected on different days between the PlanetLab node *planetlab1.nbgisp.com* and the PalnetLab node *planetlab1.gti-dsl.nodes.planet-lab.org*. These new data sets are taken as three new paths, Path A_{new}, Path B_{new}, and Path C_{new}. Their CLRs are 7.5%, 18.4%, and 5.4%, respectively. The trace-files of paths A, B, and C are called "original trace-files" in this research, and the trace-files of paths A_{new}, B_{new}, and C_{new} are called "new trace-files".

a. Prediction results on the original trace-files

For paths A, B, and C, the five trace-files of a path, which are not used for the development of the predictors, are used as the validation sets. These trace-files are used to compare the prediction results of the predictors in MSE sense. The prediction results of CLR signals, delay signals, and accumulation signals are given in Table XVIII, XIX, and XX, respectively. Each resulting MSE is calculated based on the real signal and the average prediction results of the four predictors of same model structure for the same path and same type of information signal. The MSE results of the predictions of the first four trace-files are also listed in these tables. The prediction results of SP predictors are also included in these tables. There are two prediction results from ad hoc predictors for predicting the information signals of Path A. One result is from the predictions for switching between Path A and Path B. The other result is from the predictions for switching between Path A and Path C.

These results show that the prediction difficulties of the three types of signals are very different. The resulting MSE values for the prediction of accumulation signals are the smallest. The resulting MSEs of delay signals are the second smallest. And

the resulting MSEs of CLR signals are the largest. These results indicate that the accumulation signals are easier to predict than the delay signals, and the delay signals are easier to predict than the CLR signals. An example of the prediction results are given in Fig. 42. The plots show that for all these predictions, there are clear time shifts in the prediction results compared to the real signal. None of the predictors is giving a satisfactory result.

On the average the ranking of the predictors for the prediction of the CLR signals from the best to the worst is *RBF, NAR, AR, ad hoc, SP*; the ranking of the predictors for the prediction of the delay signals from the best to the worst is *NAR, RBF, AR, ad hoc, SP*; the ranking of the predictors for the prediction of the accumulation signals from the best to the worst is *AR, NAR, ad hoc, SP, RBF*.

b. Prediction results on the new trace-files

To validate the above results, the predictors developed for the original trace-files (paths A, B, and C) are tested with the new trace-files (paths A_{new}, B_{new}, and C_{new}). The prediction results of CLR signals, delay signals, and accumulation signals are given in Table XXI, XXII, XXIII, respectively.

Again, the results show that the resulting MSEs for the prediction of accumulation signals are smaller than those of delay signals. The resulting MSEs of delay signals are smaller than those of CLR signals. On the average, the ranking of the predictors for the prediction of the CLR signals from the best to the worst is *AR, RBF, NAR, ad hoc, SP*; the ranking of the predictors for the prediction of the delay signals from the best to the worst is *RBF, NAR, AR, ad hoc, SP*; the ranking of the predictors for the prediction of the accumulation signals from the best to the worst is *AR, NAR, ad hoc, SP, RBF*.

Compared to the ranking of the predictors based on the original trace-files, the

Table XVIII. Prediction results on the CLR signals of original trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg. of
		1	2	3	4	5	6	7	8	9	$5 \sim 9$(%)[1]
CLR	Path A										
	SP	42.66	68.32	71.17	70.43	79.17	62.73	65.39	22.54	24.21	50.81
	AR	35.29	55.76	57.10	55.82	62.59	50.14	52.32	21.36	22.36	41.75
	NAR	36.19	54.73	55.56	54.68	60.53	49.38	51.76	23.13	23.89	41.74
	RBF	33.87	54.19	54.29	54.02	60.34	47.68	51.14	19.77	20.78	39.94
	Adhoc_{AB}[2]	35.59	57.36	59.28	58.24	65.22	52.32	54.37	19.57	20.91	42.48
	Adhoc_{AC}[3]	35.13	56.76	58.47	57.31	64.08	51.64	53.60	19.59	20.90	41.96
	Path B										
	SP	57.71	77.39	63.57	60.27	74.39	85.44	75.52	78.34	64.40	75.62
	AR	44.93	59.02	49.37	48.06	57.60	64.83	58.43	60.61	49.34	58.16
	NAR	43.16	56.81	46.93	47.18	55.63	62.44	56.23	58.08	48.42	56.16
	RBF	42.57	56.26	45.57	46.51	54.99	60.91	54.78	57.16	47.65	55.10
	Adhoc_{AB}	47.35	63.06	52.40	49.63	60.73	69.13	61.62	63.78	52.42	61.54
	Path C										
	SP	28.85	86.34	79.00	72.87	55.26	68.61	44.66	58.62	67.22	58.87
	AR	26.59	68.60	62.79	58.48	47.01	55.31	39.13	48.63	53.56	48.73
	NAR	35.98	66.24	62.13	58.37	48.32	55.04	43.16	51.50	56.52	50.91
	RBF	24.09	65.65	60.24	56.51	46.22	54.89	38.31	47.57	53.76	48.15
	Adhoc_{AC}	23.89	69.77	64.40	59.75	46.21	56.41	39.03	49.15	53.74	48.91

[1]Average MSE of trace-files No.5 through No.9.

[2]Ad hoc predictor for path switching control between Path A and Path B.

[3]Ad hoc predictor for path switching control between Path A and Path C.

Table XIX. Prediction results on the delay signals of original trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg. of
		1	2	3	4	5	6	7	8	9	$5 \sim 9$ (%)
Delay	Path A										
	SP	17.33	26.77	29.91	25.53	29.78	21.94	26.73	8.17	8.19	18.96
	AR	15.27	23.54	26.20	22.83	26.60	19.99	24.50	7.72	7.88	17.34
	NAR	15.41	23.06	25.80	22.22	25.90	19.61	23.92	7.93	7.91	17.06
	RBF	14.99	21.16	23.82	20.94	24.33	18.44	22.95	7.51	7.69	16.18
	Adhoc_{AB}	16.12	26.32	28.28	25.26	28.11	23.05	26.24	9.25	9.37	19.20
	Adhoc_{AC}	15.30	24.04	26.53	22.97	26.39	20.15	24.00	7.72	7.76	17.20
	Path B										
	SP	24.16	31.18	20.07	32.00	35.50	37.05	32.83	36.76	31.27	34.68
	AR	21.09	26.56	17.86	27.15	30.37	31.38	28.38	31.43	26.52	29.62
	NAR	20.42	25.26	16.56	26.56	28.95	29.81	26.90	29.91	26.13	28.34
	RBF	20.15	24.74	16.19	26.05	28.25	29.26	26.64	29.11	26.06	27.86
	Adhoc_{AB}	22.80	28.58	21.51	28.75	31.71	32.49	30.17	32.49	27.60	30.89
	path C										
	SP	11.42	37.16	36.51	35.69	29.72	31.26	16.93	24.53	35.39	27.57
	AR	10.14	31.37	30.91	30.48	24.77	26.07	16.04	21.62	30.19	23.74
	NAR	10.35	29.96	29.35	29.47	24.39	25.77	15.33	20.96	29.43	23.18
	RBF	15.25	26.12	27.25	28.26	26.75	24.49	18.70	22.59	31.91	24.89
	Adhoc_{AC}	10.21	32.37	31.95	31.28	26.26	27.51	16.24	22.19	30.42	24.53

Table XX. Prediction results on the accumulation signals of original trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg. of
		1	2	3	4	5	6	7	8	9	$5 \sim 9$ (%)
Accum.	Path A										
	SP	8.20	15.51	19.04	13.18	18.37	7.70	14.09	4.55	4.34	9.81
	AR	7.61	14.35	17.56	12.44	17.01	7.78	13.54	4.42	4.30	9.41
	NAR	7.91	13.30	16.55	12.56	15.88	8.80	13.22	5.24	5.17	9.66
	RBF	11.70	13.57	16.46	12.39	15.79	11.47	13.63	9.18	8.79	11.77
	Adhoc_{AB}	7.83	14.80	18.10	12.67	17.46	7.67	13.53	4.43	4.21	9.46
	Adhoc_{AC}	8.15	15.42	18.92	13.10	18.25	7.67	14.00	4.53	4.32	9.75
	Path B										
	SP	10.84	17.08	10.91	17.62	21.15	21.36	19.13	20.16	18.50	20.06
	AR	10.39	15.52	10.16	16.02	19.25	19.44	17.60	18.53	16.67	18.30
	NAR	12.19	14.57	9.77	15.35	18.08	18.16	16.44	17.21	15.90	17.16
	RBF	50.98	13.98	9.16	18.73	17.98	17.51	16.05	17.00	18.41	17.39
	Adhoc_{AB}	10.51	16.27	10.53	16.86	20.10	20.26	18.22	19.17	17.63	19.08
	Path C										
	SP	5.07	18.12	15.26	18.74	19.82	15.25	9.02	13.77	17.14	15.00
	AR	4.77	15.91	14.18	16.57	16.86	13.67	8.55	12.41	15.52	13.40
	NAR	8.48	15.15	20.12	15.94	16.15	13.72	9.78	12.02	19.92	14.32
	RBF	32.83	13.01	25.06	15.78	19.29	15.19	19.52	13.29	28.57	19.17
	Adhoc_{AC}	5.04	18.00	15.17	18.61	19.68	15.15	8.97	13.69	17.03	14.91

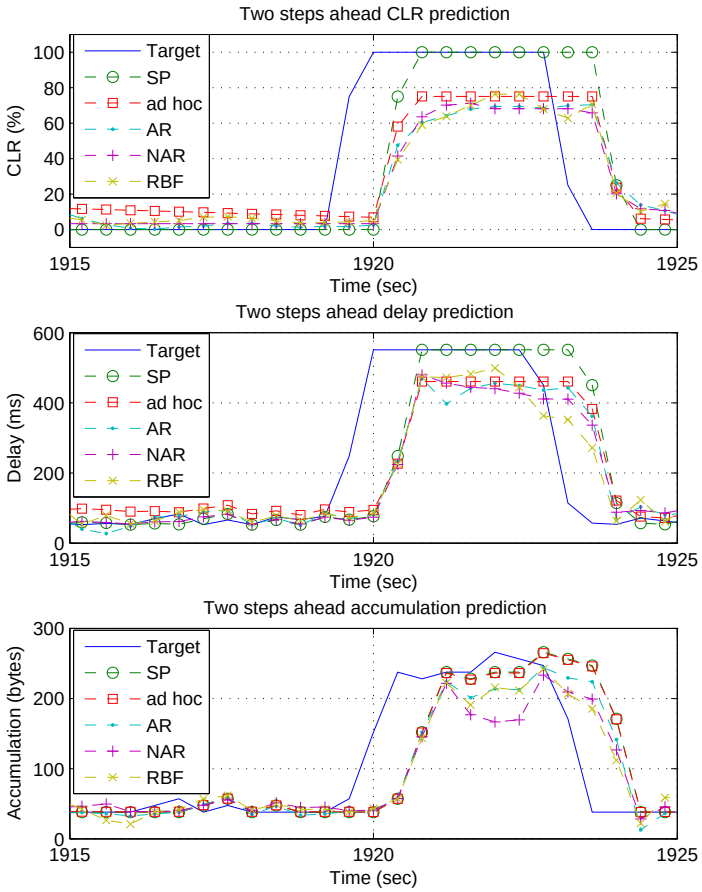

Fig. 42. An example of two-step-ahead prediction results.

ranking of AR, NAR, and RBF predictors has changed. The results show that the nonlinear predictors are not always better than the linear predictors. But except for the RBF predictor for accumulation signals, AR, NAR, and RBF predictors are always better than ad hoc predictors, and ad hoc predictors are better than SP predictors.

Table XXI. Prediction results on the CLR signals of new trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg.[1]
		1	2	3	4	5	6	7	8	9	(%)
CLR	$PathA_{new}$										
	SP	27.91	44.62	56.89	54.38	67.77	55.90	62.67	50.96	60.69	53.53
	AR	25.37	38.59	47.87	47.21	55.27	46.34	52.48	44.17	49.16	45.16
	NAR	26.74	38.52	47.12	46.59	54.27	46.57	51.55	43.90	49.00	44.92
	RBF	23.94	37.45	46.58	46.83	54.42	45.77	50.99	43.38	48.24	44.18
	$Adhoc_{AB}$	24.21	38.39	48.50	47.59	56.85	47.96	53.76	44.44	51.17	45.87
	$Adhoc_{AC}$	24.24	38.32	48.26	47.79	56.23	47.81	53.59	44.56	50.71	45.72
	$PathB_{new}$										
	SP	52.15	61.31	28.24	50.10	82.06	34.70	20.71	26.28	51.05	45.18
	AR	42.08	49.35	24.36	40.13	63.82	29.82	19.65	22.87	43.28	37.26
	NAR	45.76	51.55	24.58	41.01	65.88	29.80	19.60	23.40	47.57	38.79
	RBF	48.28	53.17	25.02	42.18	68.00	30.03	19.50	23.83	49.78	39.98
	$Adhoc_{AB}$	43.20	50.76	24.05	42.79	66.73	29.65	19.24	22.37	43.41	38.02
	$PathC_{new}$										
	SP	53.01	56.30	48.40	63.92	59.82	51.00	52.86	54.38	55.52	55.02
	AR	46.70	49.19	44.21	54.15	49.87	46.06	46.41	47.19	48.47	48.03
	NAR	46.19	48.65	44.86	53.34	50.77	45.02	45.62	47.78	48.55	47.86
	RBF	46.58	47.74	43.19	52.60	51.70	44.96	45.93	46.16	47.43	47.36
	$Adhoc_{AC}$	47.02	49.40	43.87	54.50	49.16	45.77	46.79	47.50	48.46	48.05

[1] Average MSE of trace-files No.1 through No.9.

5. Section summary

In this section, five types of predictors: SP, AR, NAR, RBF, and ad hoc, are developed for three types of information signals: CLR, delay, and accumulation. It turns out that the two-step-ahead prediction results in terms of MSE for accumulation signals are smaller than those of delay signals. The two-step-ahead prediction results in

Table XXII. Prediction results on the delay signals of new trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg.
		1	2	3	4	5	6	7	8	9	(%)
Delay	$PathA_{new}$										
	SP	9.53	18.32	23.07	21.27	31.28	24.73	26.50	22.57	25.19	22.50
	AR	8.92	16.93	21.23	19.89	29.17	22.80	24.11	20.95	23.69	20.85
	NAR	11.10	16.53	20.53	18.76	27.84	21.60	23.11	20.55	22.85	20.32
	RBF	9.28	15.71	19.63	18.31	26.46	20.77	21.97	19.51	21.69	19.26
	$Adhoc_{AB}$	10.97	20.16	24.08	24.65	30.01	26.19	27.73	24.89	26.80	23.94
	$Adhoc_{AC}$	9.05	17.10	21.14	20.22	27.87	22.78	24.31	21.09	23.23	20.75
	$PathB_{new}$										
	SP	24.64	35.69	10.67	24.16	40.32	19.09	11.41	11.71	19.25	21.88
	AR	22.04	30.74	10.12	21.02	35.08	17.40	10.79	10.85	17.91	19.55
	NAR	20.31	28.89	10.10	20.26	33.24	17.63	11.24	10.74	17.90	18.92
	RBF	19.59	28.51	10.28	19.76	32.29	17.84	11.57	10.99	18.53	18.82
	$Adhoc_{AB}$	22.71	30.27	11.95	24.61	34.01	19.28	14.35	12.57	20.21	21.11
	$PathC_{new}$										
	SP	23.40	24.51	23.09	26.54	29.07	23.54	24.25	26.36	24.94	25.08
	AR	20.23	21.43	19.46	23.23	25.32	20.15	21.67	23.36	22.40	21.92
	NAR	18.66	19.86	17.65	20.88	24.58	18.14	20.77	23.06	21.22	20.53
	RBF	20.44	17.59	15.73	18.22	27.98	16.44	21.03	24.28	20.62	20.26
	$Adhoc_{AC}$	21.89	22.80	21.65	24.29	25.76	22.03	22.56	24.26	23.07	23.15

Table XXIII. Prediction results on the accumulation signals of new trace-files.

Signal	Method	MSE of predicting trace-file (%)									Avg.
		1	2	3	4	5	6	7	8	9	(%)
Accum.	$PathA_{new}$										
	SP	5.11	8.94	13.31	11.57	17.91	13.30	13.64	13.81	13.57	12.35
	AR	5.04	8.67	12.45	10.82	16.59	12.39	12.73	13.24	12.92	11.65
	NAR	5.40	9.27	12.06	10.96	15.65	11.79	11.84	14.35	12.39	11.52
	RBF	8.06	9.73	12.09	10.87	15.53	11.56	12.11	13.70	12.64	11.81
	$Adhoc_{AB}$	4.95	8.70	12.73	11.18	17.03	12.74	13.02	13.24	13.10	11.85
	$Adhoc_{AC}$	5.09	8.89	13.23	11.50	17.79	13.22	13.55	13.72	13.49	12.28
	$PathB_{new}$										
	SP	13.24	23.50	9.95	13.36	15.51	9.53	4.93	6.61	12.15	12.09
	AR	13.04	22.70	9.26	13.03	15.04	9.02	4.93	6.32	11.44	11.64
	NAR	12.75	21.65	10.37	12.96	18.18	13.27	10.84	8.26	11.16	13.27
	RBF	20.70	23.52	10.30	16.78	76.07	83.32	124.40	32.65	12.77	44.50
	$Adhoc_{AB}$	12.72	22.25	9.53	12.89	14.97	9.34	4.98	6.44	11.69	11.64
	$PathC_{new}$										
	SP	11.42	12.24	11.23	13.25	17.44	11.31	12.05	15.80	13.03	13.09
	AR	10.33	11.11	10.20	11.98	15.20	10.29	10.93	14.12	11.88	11.78
	NAR	10.11	10.77	9.92	11.51	14.86	9.97	10.81	13.67	11.55	11.46
	RBF	25.55	26.91	17.47	24.41	94.31	100.38	143.37	40.52	24.24	55.24
	$Adhoc_{AC}$	11.35	12.16	11.16	13.16	17.33	11.24	11.98	15.70	12.95	13.00

terms of MSE for delay signals are smaller than those of CLR signals. The ranking of the predictors for each type of information signal is different. On the average, the AR, NAR, and RBF predictors are better than ad hoc predictors, and the ad hoc predictors are better than SP predictors, except that the prediction results of RBF predictors for accumulation signals are worse than the prediction results of ad hoc predictors and SPs for accumulation signals. This might be because that the accumulation signals of the training set does not covered the whole range of all the possible accumulations values.

D. Chapter Summary

In the chapter, first the requirements on the prediction horizon is discussed. The relation between the prediction step size, control step size, decision step size, and the minimum required prediction horizon is discussed. It proves that a minimum prediction horizon of two and only two steps ahead is needed in this study.

Second, the impact of prediction/switching step size (interval) T_p on the predictive path switching control results is investigated. The results show that smaller switching interval gives better path switching control results.

Five types of predictors, SP, AR, NAR, RBF, and ad hoc, are developed for three types of information signals, CLR, delay, and accumulation. The two-step-ahead prediction results for accumulation signals in terms of MSE are smaller than those of delay. The two-step-ahead prediction results for delay signals are smaller than those of CLR signals. On the average, the AR, NAR, and RBF predictors are better than the ad hoc predictors, and the ad hoc predictors are better than the SP predictors, except that the RBF predictors for accumulation signals are worse than ad hoc predictors and SPs for accumulation signals.

CHAPTER VIII

RESULTS FROM PREDICTIVE PATH SWITCHING CONTROL

In this chapter, firstly, the control logic of the controller developed is given. Secondly, the control results based on different predictors are presented. Thirdly, a voting based predictive path switching control method is proposed and investigated. Finally, a new predictor evaluation criterion other than MSE is discussed.

A. Predictive Path Switching Control Logic

Without losing generality, consider predictive path switching control between Path A and Path B as an example, given the four two-step-ahead predictors

$$\{p_1(\cdot), p_2(\cdot), p_3(\cdot), p_4(\cdot)\}$$

of any chosen variety. The proposed control logic is as follows:

1. At time k, send probing packets to both Path A and Path B.

2. Calculate the information signals $y_A(k-2)$, $y_B(k-2)$ from the available measurements which carry information from probing packets sent.

3. Calculate the two-step-ahead predictions:
   $\hat{y}_A(k|k-2) = \sum_{i=1}^{4} p_i[y_A(k-2), y_A(k-3), \ldots]$;
   $\hat{y}_B(k|k-2) = \sum_{i=1}^{4} p_i[y_B(k-2), y_B(k-3), \ldots]$.

4. Considering the information signal carry path congestion information,
   if $\hat{y}_A(k|k-2) < \hat{y}_B(k|k-2)$
   Transmit packets over Path A only.
   else if $\hat{y}_A(k|k-2) > \hat{y}_B(k|k-2)$

Transmit packets over Path B only.

else if $\hat{y}_A(k|k-2) == \hat{y}_B(k|k-2)$

Do not switch, but transmit the packets over the path used in the previous time step.

end if

5. Go back to step 1

B. Predictive Path Switching Control Results

1. Results of the original trace-files

The results of predictive path switching control based on different types of predictors in terms of CLR are discussed below. Table XXIV gives the results of predictive path switching control between Path A and Path B. Table XXV gives the results of predictive path switching control between Path A and Path C. The predictors used in predictive path switching controls are: SPs for CLR signals ($\mathrm{SP_{CLR}}$), AR predictors for CLR signals ($\mathrm{AR_{CLR}}$), NAR predictors for CLR signals ($\mathrm{AAR_{CLR}}$), RBF predictors for CLR signals ($\mathrm{RBF_{CLR}}$), ad hoc predictors for CLR signals ($\mathrm{Adhoc_{CLR}}$), SPs for delay signals ($\mathrm{SP_{delay}}$), AR predictors for delay signals ($\mathrm{AR_{delay}}$), NAR predictors for delay signals ($\mathrm{AAR_{delay}}$), RBF predictors for delay signals ($\mathrm{RBF_{delay}}$), ad hoc predictors for delay signals ($\mathrm{Adhoc_{delay}}$), SPs for accumulation signals ($\mathrm{SP_{CLR}}$), AR predictors for accumulation signals ($\mathrm{AR_{accum}}$), NAR predictors for accumulation signals ($\mathrm{AAR_{accum}}$), RBF predictors for accumulation signals ($\mathrm{RBF_{accum}}$), and ad hoc predictors for accumulation signals ($\mathrm{Adhoc_{accum}}$). The resulting CLRs of every trace-files for each path pair and the mean over all the trace-files for each path pair are given in these two tables. For comparison, the CLRs of no switching and transmitting VoIP packets over only one path are also included. The CLRs of ideal case path

switching control are included as well. The top five methods, excluding the ideal case switching control, for switching between each pair of paths are boxed.

The results show that although the predictive path switching controls with two-step-ahead predictors have CLR as high as two to three times that of the ideal case path switching control, they are always better than the no switching methods. For switching between Path A and Path B, the best predictive path switching control has reduced the resulting CLRs to about half of the resulting CLRs of the best no switching method. For switching between Path A and Path C, the best predictive path switching control has reduced the resulting CLRs to about one third of the resulting CLRs of the best no switching method.

Checking the ranking of the predictors according to their resulting predictive path switching control results in terms of CLR for each trace-file, it is seen that there is no one type of predictor that always gives better predictive path switching control results than all others, at all time. On the average, for switching between Path A and Path B, NAR_{CLR} gives the best predictive path switching control results. The next best four types of predictors are RBF_{CLR}, $\text{RBF}_{\text{delay}}$, $\text{NAR}_{\text{delay}}$, and $\text{Adhoc}_{\text{CLR}}$. This shows that for switching between Path A and Path B, CLR signal based predictors give better results. For switching between Path A and Path C, on the average, the top five predictors are: $\text{Adhoc}_{\text{delay}}$, SP_{delay}, $\text{NAR}_{\text{delay}}$, AR_{delay}, $\text{RBF}_{\text{delay}}$. This shows that for switching between Path A and Path C, delay based predictors are better. So there is no universally acceptable predictor or even information signal to use.

Note that in the previous chapter (Chapter VII) on predictors, signal accumulation signals show the smallest MSE, but when used for predictive path switching control, these predictors do not appear on the top five. Another observation is that when comparing the prediction results, ad hoc predictors and SPs are always worse than the other predictors, except the RBF predictors for accumulation signals. But

Table XXIV. Predictive path switching results between Path A and Path B in terms of CLR.

Method	CLR of different controls on trace-file (%)									Avg.[1]	The top
	1	2	3	4	5	6	7	8	9	(%)	five[2]
Path A[3]	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	
Path B[4]	25.21	15.59	18.54	26.23	19.53	15.75	18.83	18.47	25.35	20.39	
CLR											
Ideal[5]	4.40	1.08	1.42	2.41	1.62	1.95	1.98	2.79	2.23	2.21	
SP	9.11	4.48	5.21	6.85	5.83	6.07	6.31	5.56	5.02	6.05	
AR	8.84	4.18	5.08	6.18	5.69	5.83	6.15	5.53	4.79	5.81	
NAR	8.59	3.36	4.66	5.87	5.01	5.44	5.74	4.84	4.43	5.33	1
RBF	8.60	3.43	4.57	5.89	5.14	5.38	5.74	4.85	4.38	5.33	2
Adhoc	8.95	3.80	5.03	6.07	5.25	5.54	6.20	5.25	4.72	5.64	5
Delay											
SP	9.52	5.47	5.61	7.37	6.80	6.65	7.33	7.11	6.73	6.96	
AR	8.94	4.78	5.07	6.32	5.81	6.36	6.13	6.10	5.28	6.09	
NAR	8.58	3.64	4.60	6.04	5.38	5.54	5.57	5.04	4.57	5.44	4
RBF	8.42	3.66	4.50	6.16	5.28	5.44	5.62	5.06	4.74	5.43	3
Adhoc	9.37	4.84	5.46	6.25	5.71	6.27	6.27	5.68	5.15	6.11	
Accum.[6]											
SP	9.90	4.76	5.39	6.96	6.42	6.04	6.27	5.70	5.50	6.33	
AR	9.44	4.68	5.16	7.12	6.15	5.99	6.64	5.94	5.75	6.32	
NAR	9.08	4.01	4.84	6.30	5.67	5.59	5.86	5.32	4.96	5.74	
RBF	9.49	4.08	5.02	6.71	5.84	5.69	6.04	5.48	5.38	5.97	
Adhoc	9.27	4.25	5.19	6.28	5.65	5.67	6.14	5.52	5.11	5.90	

[1]Average CLR of the control results of the nine trace-files.

[2]The top five controls other than the ideal case path switching control.

[3]No switching and transmitting VoIP packets only over Path A.

[4]No switching and transmitting VoIP packets only over Path B.

[5]Ideal case path switching control.

[6]Accumulation.

Table XXV. Predictive path switching results between Path A and Path C in terms of CLR.

Method	CLR of different controls on trace-file (%)									Avg.	The top
	1	2	3	4	5	6	7	8	9	(%)	five
Path A	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89	
Path C[3]	17.45	4.76	7.37	7.14	9.42	5.02	8.66	6.15	14.50	8.94	
CLR											
Ideal	2.44	0.33	0.60	0.68	0.46	0.40	1.03	2.18	1.09	1.02	
SP	4.81	2.55	3.52	3.27	3.05	1.82	3.27	3.77	3.25	3.26	
AR	5.16	2.43	3.67	3.40	3.14	2.09	3.28	3.72	3.59	3.39	
NAR	5.22	2.48	3.59	3.21	2.90	1.98	3.32	3.90	5.12	3.53	
RBF	5.16	2.34	3.45	3.15	3.10	2.53	3.34	3.94	4.30	3.48	
Adhoc	5.30	2.54	3.40	3.22	3.03	1.97	3.21	3.61	3.16	3.27	
Delay											
SP	4.70	1.95	3.01	2.91	2.71	1.82	2.87	3.57	4.07	3.06	2
AR	4.82	2.10	3.16	3.19	2.50	1.89	3.12	3.47	3.42	3.08	4
NAR	4.83	2.06	3.10	2.92	2.61	2.06	3.01	3.49	3.57	3.07	3
RBF	4.69	2.06	3.06	3.03	2.74	2.09	2.93	3.55	4.38	3.17	5
ad hoc	4.72	2.12	3.01	3.00	2.65	1.64	2.96	3.49	3.21	2.98	1
Accum.											
SP	5.05	2.47	3.19	3.63	3.16	2.25	3.14	3.96	3.28	3.35	
AR	5.55	2.79	3.62	3.91	3.48	2.75	3.87	3.93	3.37	3.70	
NAR	5.14	2.68	3.53	3.65	3.06	2.35	3.52	3.75	3.77	3.50	
RBF	5.84	2.72	3.83	3.75	3.63	2.86	3.85	3.94	4.10	3.84	
Adhoc	5.18	2.63	3.35	3.15	3.20	2.08	3.33	3.68	3.11	3.30	

[3]No switching and transmitting VoIP packets only over Path C.

sometimes when these predictors are used in predictive path switching control, ad hoc predictors and SPs give better control results than other predictors. This provides a hint that MSE might not be the best criterion for evaluating the performance of predictors used in predictive path switching control.

The path switching control results for path pair AB are plotted in Fig. 43. A zoom-in of those two-step-ahead predictors is shown in Fig. 44. The results of the SP predictors for CLR signals (SP_{CLR}) and the best predictor for path pair AB switching are marked with thicker lines. The figures show that the resulting CLRs of the predictive path switching controls are lower (better) than those of no switching methods and are higher (worse) than those of the ideal case path switching control. NAR predictor for CLR signals (NAR_{CLR}) is the best for predictive path switching control between Path A and Path B, and the resulting CLR plot for NAR_{CLR} is almost always lower (better) than those of the other predictors in this case. The path switching control results of path pair AC are plotted in Fig. 45 and Fig. 46 in the same way. The figures show that for Path A and Path C, the resulting CLRs of predictive path switching control are also always lower (better) than those of no switching methods and higher (worse) than the ideal case path switching control. The ad hoc predictor for delay signals ($Adhoc_{delay}$) is the best for predictive path switching control between Path A and Path C, and the resulting CLR plot of $Adhoc_{delay}$ is almost always lower (better) than those of the other predictors in this case.

The predictive path switching control results in terms of E-model MOS are presented in Table XXVI for path pair AB and in Table XXVII for path pair AC. The ranking of the predictors according to their predictive path switching control results in terms of E-model MOS is the same as their ranking according to their control results in terms of CLR. The top five predictors for predictive path switching control between Path A and Path B are NAR_{CLR}, RBF_{CLR}, RBF_{delay}, NAR_{delay}, and $Adhoc_{CLR}$. The

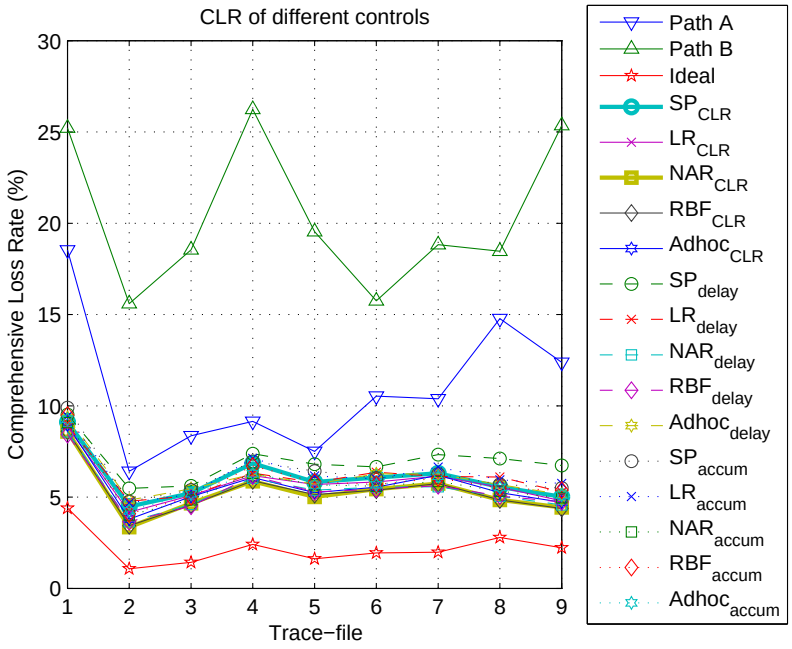

Fig. 43. Results of predictive path switching with different predictors between Path A and Path B in terms of CLR. $\mathrm{SP_{CLR}}$ and the best method, $\mathrm{NAR_{CLR}}$, are marked with thicker lines.

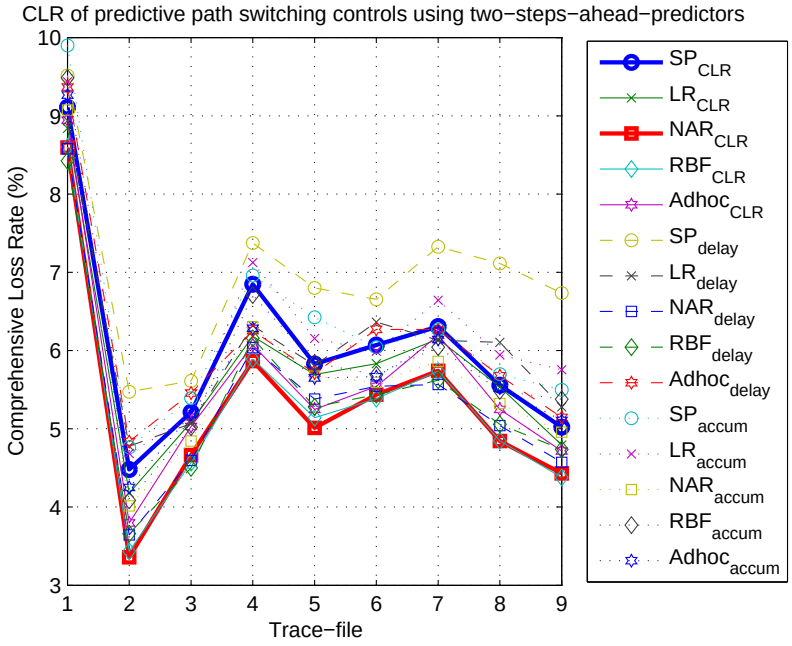

Fig. 44. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A and Path B in terms of CLR. $\mathrm{SP_{CLR}}$ and the best method, $\mathrm{NAR_{CLR}}$, are marked with thicker lines.

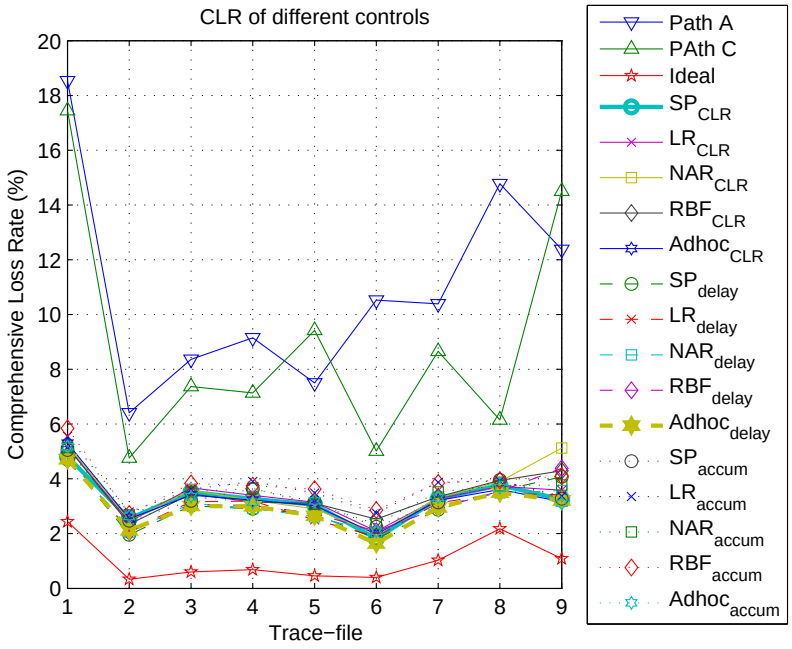

Fig. 45. Results of predictive path switching with different predictors between Path A and Path C in terms of CLR. $\mathrm{SP}_{\mathrm{CLR}}$ and the best method, $\mathrm{Adhoc}_{\mathrm{delay}}$, are marked with thicker lines.

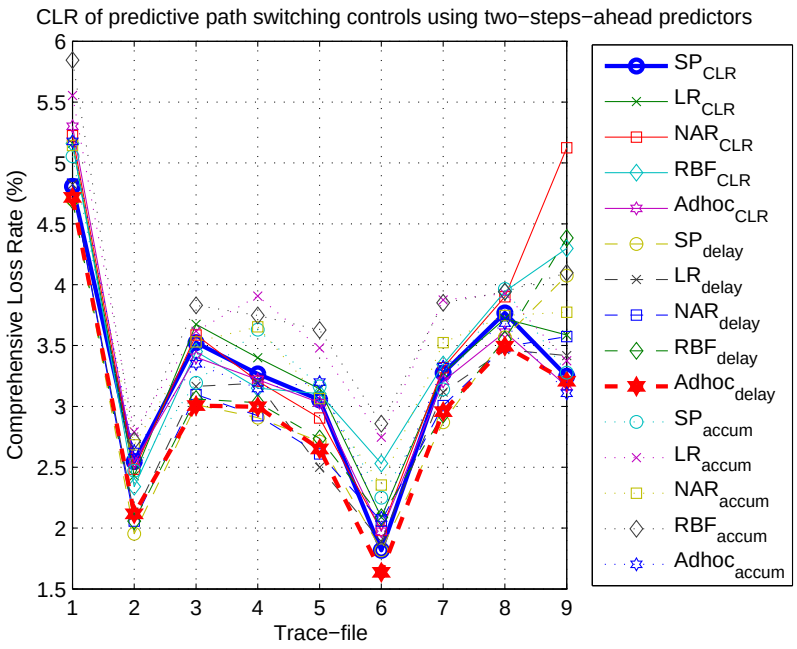

Fig. 46. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A and Path C in terms of CLR. $\mathrm{SP}_{\mathrm{CLR}}$ and the best method, $\mathrm{Adhoc}_{\mathrm{delay}}$, are marked with thicker lines.

top five predictors for predictive path switching control between Path A and Path C are $\text{Adhoc}_{\text{delay}}$, SP_{delay}, $\text{NAR}_{\text{delay}}$, AR_{delay}, and $\text{RBF}_{\text{delay}}$. No one single predictor is the best for predictive path switching control all the time, and the ranking of the predictors according to their predictive path switching control results in terms of MOS is not the same as the ranking of the predictors according to their signal prediction results in terms of MSE for each path pair.

The plots for path pair AB are in Fig. 47 and Fig. 48. The plots show that the resulting E-model MOS of the predictive path switching controls are higher (better) than those of no switching methods and lower (worse) than those of the ideal case path switching control. The NAR_{CLR} gives the best results in terms of MOS for predictive path switching control between Path A and Path B. The resulting MOS of NAR_{CLR} is almost always higher (better) than those of other predictors. The plots for path pair AC are in Fig. 49 and Fig. 50. The plots show that the resulting E-model MOS of the predictive path switching controls are higher (better) than those of no switching methods and lower (worse) than those of the ideal case path switching control. The $\text{Adhoc}_{\text{delay}}$ gives the best results in terms of MOS for predictive path switching control between Path A and Path C. The resulting MOS for $\text{Adhoc}_{\text{delay}}$ is almost always higher (better) than the MOS for other predictors.

Table XXVI. Predictive path switching results between Path A and Path B in terms of E-model MOS.

Method	E-model MOS of different controls on trace-file									Avg.[1]	The top five
	1	2	3	4	5	6	7	8	9		
Path A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	
Path B	0.99	1.43	1.19	0.99	1.19	1.46	1.22	1.26	1.00	1.19	
CLR											
Ideal	2.88	3.51	3.44	3.24	3.40	3.33	3.32	3.17	3.28	3.28	
SP	2.14	2.86	2.74	2.45	2.63	2.60	2.56	2.69	2.77	2.60	
AR	2.17	2.91	2.76	2.55	2.64	2.63	2.58	2.69	2.81	2.64	
NAR	2.19	3.06	2.82	2.60	2.76	2.69	2.64	2.79	2.86	2.71	1
RBF	2.19	3.04	2.84	2.59	2.73	2.70	2.64	2.80	2.88	2.71	2
Adhoc	2.16	2.98	2.77	2.56	2.72	2.68	2.56	2.73	2.82	2.66	5
Delay											
SP	2.10	2.70	2.68	2.38	2.49	2.53	2.42	2.45	2.50	2.47	
AR	2.17	2.81	2.77	2.54	2.64	2.56	2.60	2.60	2.73	2.60	
NAR	2.20	3.01	2.84	2.58	2.70	2.68	2.68	2.77	2.85	2.70	4
RBF	2.23	3.00	2.86	2.56	2.72	2.70	2.67	2.77	2.82	2.70	3
Adhoc	2.11	2.80	2.71	2.54	2.65	2.58	2.57	2.67	2.75	2.60	
Accum.											
SP	2.04	2.82	2.72	2.43	2.54	2.61	2.57	2.67	2.69	2.56	
AR	2.09	2.83	2.75	2.41	2.57	2.61	2.51	2.62	2.64	2.56	
NAR	2.14	2.94	2.80	2.53	2.65	2.67	2.63	2.72	2.78	2.65	
RBF	2.05	2.93	2.77	2.47	2.62	2.65	2.60	2.69	2.70	2.61	
Adhoc	2.12	2.90	2.75	2.54	2.65	2.67	2.59	2.69	2.75	2.63	

[1]Average MOS of the control results of the nine trace-files.

Table XXVII. Predictive path switching results between Path A and Path C in terms of E-model MOS.

Method	MOS of different controls on trace-file									Avg.	The top five
	1	2	3	4	5	6	7	8	9		
Path A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82	
Path C	1.10	2.82	2.36	2.41	2.03	2.77	2.10	2.56	1.52	2.19	
CLR											
Ideal	3.24	3.67	3.61	3.59	3.64	3.65	3.52	3.28	3.51	3.52	
SP	2.81	3.21	3.03	3.08	3.12	3.36	3.08	2.98	3.08	3.08	
AR	2.73	3.24	3.00	3.05	3.10	3.30	3.07	2.99	3.02	3.06	
NAR	2.69	3.23	3.02	3.09	3.15	3.33	3.07	2.96	2.76	3.03	
RBF	2.74	3.25	3.04	3.10	3.11	3.22	3.06	2.95	2.89	3.04	
Adhoc	2.73	3.21	3.05	3.09	3.12	3.33	3.09	3.01	3.09	3.08	
Delay											
SP	2.83	3.33	3.13	3.15	3.18	3.36	3.15	3.02	2.94	3.12	2
AR	2.81	3.30	3.10	3.09	3.22	3.34	3.11	3.04	3.05	3.12	4
NAR	2.81	3.31	3.11	3.14	3.20	3.31	3.13	3.03	3.02	3.12	3
RBF	2.83	3.31	3.12	3.12	3.18	3.30	3.14	3.02	2.88	3.10	5
ad hoc	2.82	3.30	3.13	3.13	3.19	3.40	3.14	3.03	3.09	3.14	1
Accum.											
SP	2.77	3.23	3.09	3.01	3.10	3.27	3.10	2.95	3.07	3.07	
AR	2.67	3.16	3.01	2.96	3.04	3.17	2.97	2.95	3.05	3.00	
NAR	2.75	3.19	3.03	3.01	3.12	3.25	3.03	2.99	2.99	3.04	
RBF	2.61	3.18	2.98	2.99	3.01	3.15	2.97	2.95	2.93	2.97	
Adhoc	2.75	3.20	3.06	3.10	3.09	3.30	3.07	3.00	3.10	3.07	

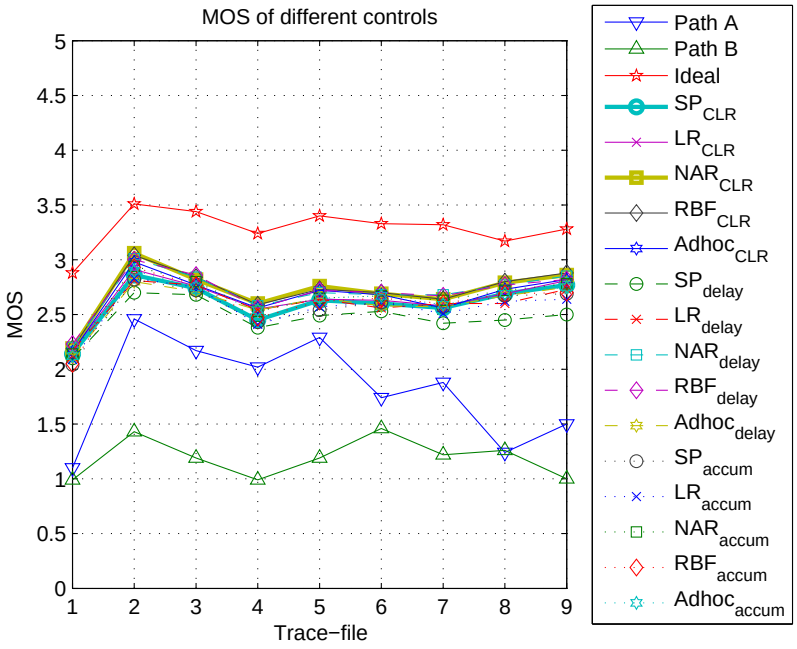

Fig. 47. Results of predictive path switching with different predictors between Path A and Path B in terms of E-model MOS. $\mathrm{SP_{CLR}}$ and the best method, $\mathrm{NAR_{CLR}}$, are marked with thicker lines.

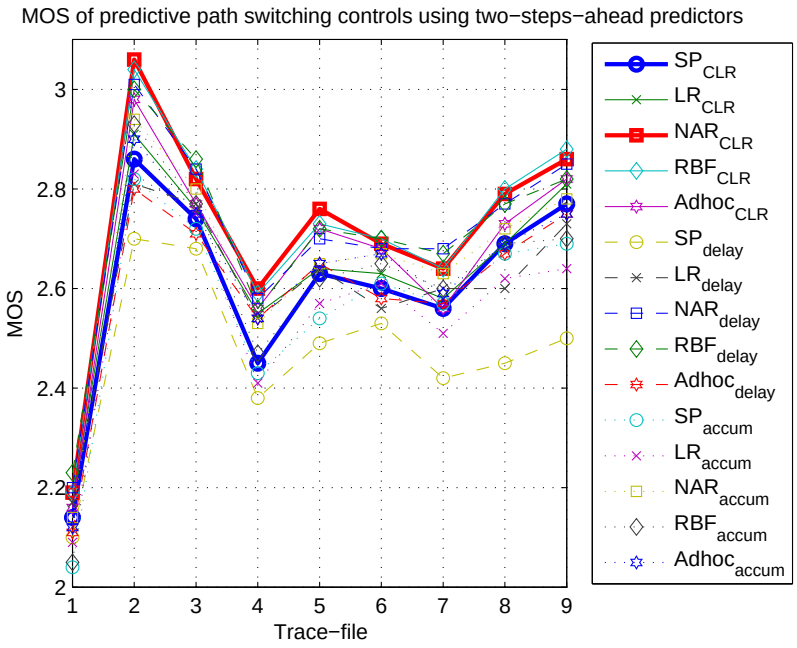

Fig. 48. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A and Path B in terms of E-model MOS. $\mathrm{SP_{CLR}}$ and the best method, $\mathrm{NAR_{CLR}}$, are marked with thicker lines.

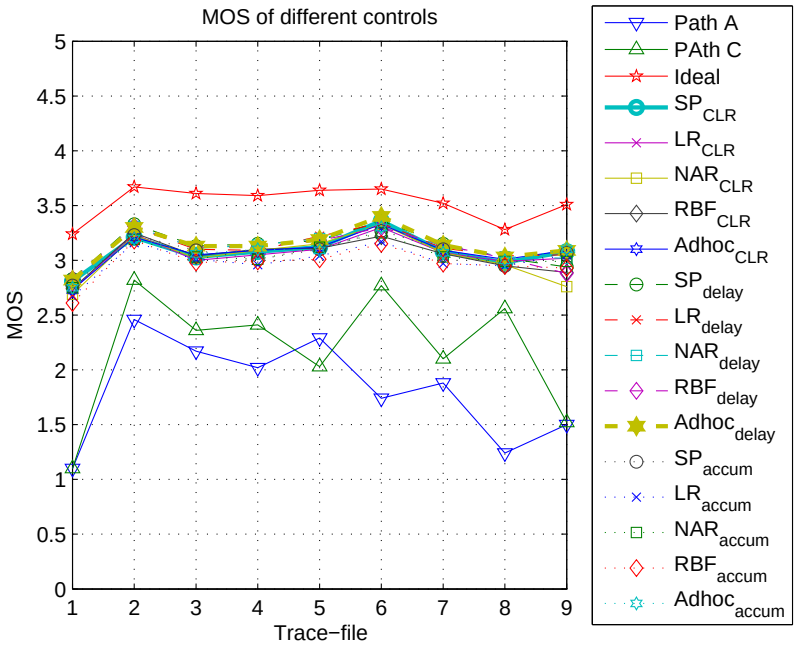

Fig. 49. Results of predictive path switching with different predictors between Path A and Path C in terms of E-model MOS. $\mathrm{SP}_{\mathrm{CLR}}$ and the best method, $\mathrm{Adhoc}_{\mathrm{delay}}$, are marked with thicker lines.

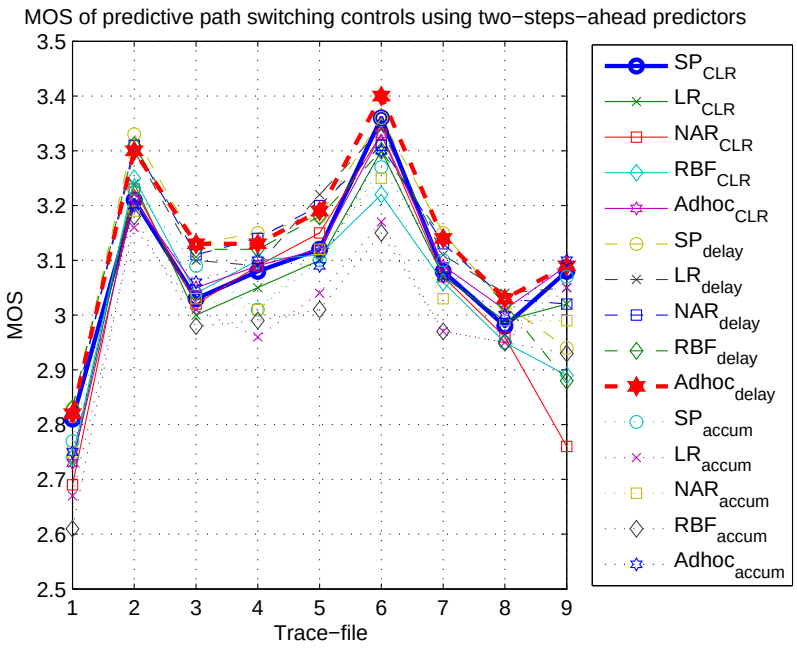

Fig. 50. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A and Path C in terms of E-model MOS. $\mathrm{SP}_{\mathrm{CLR}}$ and the best method, $\mathrm{Adhoc}_{\mathrm{delay}}$, are marked with thicker lines.

2. Results of the new trace-files

The predictors developed with original paths A, B and C, are used in predictive path switching control for the new paths A_{new}, B_{new}, and C_{new}. Table XXVIII and Table XXIX give the results of predictive path switching control between paths A_{new}, B_{new}, and paths A_{new}, C_{new}, respectively. The no switching methods and ideal case path switching control results are also included. The top five methods, other than the ideal case path switching control, for predictive path switching control between each pair of paths are boxed.

Again, the results show that the best predictive path switching control reduces the resulting CLRs to half or even one third of the resulting CLRs of no switching methods. On the average, the best five predictors for predictive path switching controls on path pair $A_{new}B_{new}$ are NAR_{delay}, RBF_{delay}, AR_{delay}, $Adhoc_{delay}$, and SP_{CLR}. The best five predictors for predictive path switching controls on path pair $A_{new}C_{new}$ are SP_{delay}, $Adhoc_{delay}$, NAR_{delay}, AR_{delay}, and RBF_{CLR}. It seems that for these two path pairs, the delay signal based predictive path switching controls are relatively better. But checking the details, it can be seen that still no one predictive path switching controller is the best at all time. For example, for the 9^{th} trace-file of path pair $A_{new}C_{new}$ the NAR_{CLR} is better than SP_{delay}. Also, the ranking of these predictors according to their predictive path switching control results in CLR sense does not match with the ranking of these predictors according to their two-step-ahead predictions in MSE sense.

The predictive path switching control results in terms of CLR and a zoom-in on the predictive path switching control results on two-step-ahead predictors of path pair $A_{new}B_{new}$ are plotted in Fig. 51 and Fig. 52. The results of the SP predictors based on CLR (SP_{CLR}), and the best methods for each path pair are also marked with

thicker lines. The figures show that the resulting CLR of the predictive path switching controls are lower (better) than those of no switching methods and higher (worse) than those of ideal case path switching control. The NAR predictor for delay signals (NAR_{delay}) is the best for predictive path switching control between Path A_{new} and Path B_{new}, and the resulting CLR plot of NAR_{delay} is almost always lower (better) than the CLRs of the other predictors in this case. The the results of path pair $A_{new}C_{new}$ are plotted in Fig. 53 and Fig. 54. The figures show that the resulting CLR of the predictive path switching controls are lower (better) than those of no switching methods and higher (worse) than those of the ideal case path switching control. The SPs using delay signals (SP_{delay}) are the best for predictive path switching control between Path A_{new} and Path C_{new}, and the resulting CLR plot of SP_{delay} is almost always lower (better) than the CLRs of the other predictors in this case.

The predictive path switching control results in terms of E-model MOS are presented in Table XXX for path pair $A_{new}B_{new}$ and in Table XXXI for path pair $A_{new}C_{new}$. The ranking of the predictors according to their predictive path switching control results are the same as given by the CLR results. The top five predictors for predictive path switching control between Path A_{new} and Path B_{new} are NAR_{delay}, RBF_{delay}, AR_{delay}, $Adhoc_{delay}$, and SP_{CLR}. The top five predictors for predictive path switching control between Path A_{new} and Path C_{new} are SP_{delay}, $Adhoc_{delay}$, NAR_{delay}, AR_{delay}, and RBF_{delay}. No one single predictor is the best for predictive path switching control all the time. The ranking of the predictors according to their predictive path switching control results in terms of MOS is not the same as the ranking of the predictors according to their signal prediction control results in terms of MSE.

The plots for path pair $A_{new}B_{new}$ are in Fig. 55 and Fig. 56. The plots show that the resulting E-model MOS of the predictive path switching controls are higher (better) than those of no switching methods and lower (worse) than those of the ideal

Table XXVIII. Predictive path switching results between Path A_{new} and Path B_{new} in terms of CLR.

Method	CLR of different controls on trace-file (%)									Avg.	The top
	1	2	3	4	5	6	7	8	9	(%)	five
Path A_{new}	8.70	8.95	7.11	5.48	7.91	6.91	5.07	6.32	11.22	7.52	
Path B_{new}	12.18	12.84	23.39	13.31	8.88	28.70	33.48	24.08	8.99	18.43	
CLR											
Ideal	1.55	0.88	2.76	0.43	0.59	2.04	1.55	1.36	1.74	1.43	
SP	2.89	2.72	4.53	2.02	3.35	3.79	3.31	2.55	4.15	3.26	5
AR	3.07	3.04	4.64	2.47	3.35	4.00	3.26	2.92	4.58	3.48	
NAR	3.21	3.28	4.63	2.20	3.57	3.60	3.09	2.82	5.21	3.51	
RBF	3.37	3.54	4.59	2.23	3.50	3.89	3.14	2.95	5.14	3.59	
ad hoc	3.07	2.84	4.62	2.25	3.46	3.98	3.27	2.47	4.19	3.35	
Delay											
SP	3.23	2.83	4.61	2.26	3.02	3.84	2.87	2.74	4.45	3.32	
AR	2.97	2.80	4.47	2.19	2.99	3.59	2.90	2.70	4.35	3.22	3
NAR	2.85	2.61	4.38	1.99	2.94	3.42	2.77	2.64	4.50	3.12	1
RBF	2.75	2.62	4.38	1.99	3.04	3.48	2.81	2.68	4.56	3.15	2
ad hoc	2.99	2.69	4.43	2.18	3.09	3.68	3.09	2.49	4.34	3.22	4
Accum.											
SP	3.06	2.81	4.58	2.21	3.25	3.89	3.37	2.70	4.35	3.36	
AR	3.46	3.88	5.30	2.83	3.88	4.62	3.46	3.57	4.73	3.97	
NAR	3.25	3.58	4.83	2.40	3.50	4.04	3.31	3.14	4.95	3.67	
RBF	3.60	4.23	5.20	2.83	4.03	6.52	9.89	4.15	5.24	5.08	
Adhoc	2.96	2.91	4.79	2.31	3.32	4.01	3.26	2.57	4.19	3.37	

Table XXIX. Predictive path switching results between Path A_{new} and Path C_{new} in terms of CLR.

Method	CLR of different controls on trace-file (%)									Avg.	The top
	1	2	3	4	5	6	7	8	9	(%)	five*
Path A_{new}	8.70	8.95	7.11	5.48	7.91	6.91	5.07	6.32	11.22	7.52	
Path C_{new}	4.24	4.24	4.16	4.58	10.88	4.13	4.74	7.04	4.89	5.43	
CLR											
Ideal	0.21	0.44	0.21	0.19	0.57	0.28	0.51	0.41	0.42	0.36	
SP	1.38	2.01	1.50	1.60	2.82	1.39	1.67	2.01	2.14	1.84	
AR	1.36	2.14	1.70	1.76	3.02	1.41	1.76	2.07	2.18	1.93	
NAR	1.33	1.79	1.28	1.75	3.58	1.39	1.85	2.10	1.87	1.88	
RBF	1.45	1.89	1.66	1.80	3.69	1.50	1.88	2.25	2.24	2.04	
Adhoc	1.34	2.00	1.63	1.73	2.78	1.49	1.83	2.11	2.02	1.88	
Delay											
SP	1.26	1.49	1.03	1.30	2.93	1.39	1.54	1.62	2.01	1.62	1
AR	1.30	1.57	1.24	1.41	3.09	1.44	1.65	1.83	2.37	1.77	4
NAR	1.25	1.57	1.07	1.48	3.02	1.25	1.55	1.58	1.96	1.64	3
RBF	1.34	1.54	1.09	1.46	3.62	1.44	1.61	2.15	2.24	1.83	5
Adhoc	1.22	1.62	1.24	1.44	2.55	1.41	1.54	1.84	1.80	1.63	2
Accum.											
SP	1.42	2.03	1.36	1.83	2.89	1.42	1.74	1.97	2.06	1.86	
AR	1.63	2.36	2.04	1.68	3.21	1.88	2.09	2.19	3.07	2.24	
NAR	1.33	1.95	1.48	1.76	3.05	1.66	1.84	2.26	2.34	1.97	
RBF	1.57	2.20	1.79	1.80	3.24	1.80	2.06	2.31	2.89	2.19	
Adhoc	1.33	1.97	1.64	1.74	2.80	1.53	1.91	2.08	1.93	1.88	

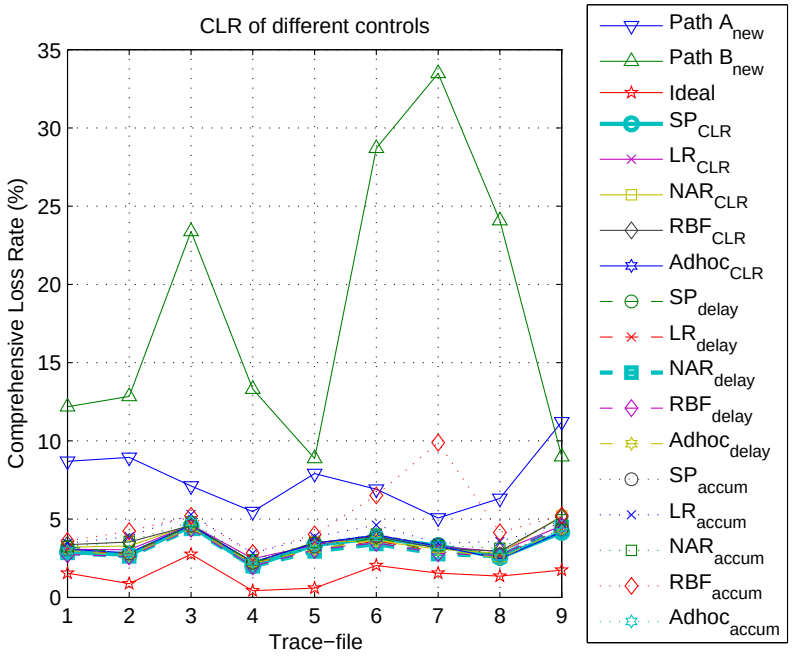

Fig. 51. Results of predictive path switching with different predictors between Path A_{new} and Path B_{new} in terms of CLR. SP_{CLR} and the best method, NAR_{delay}, are marked with thicker lines.

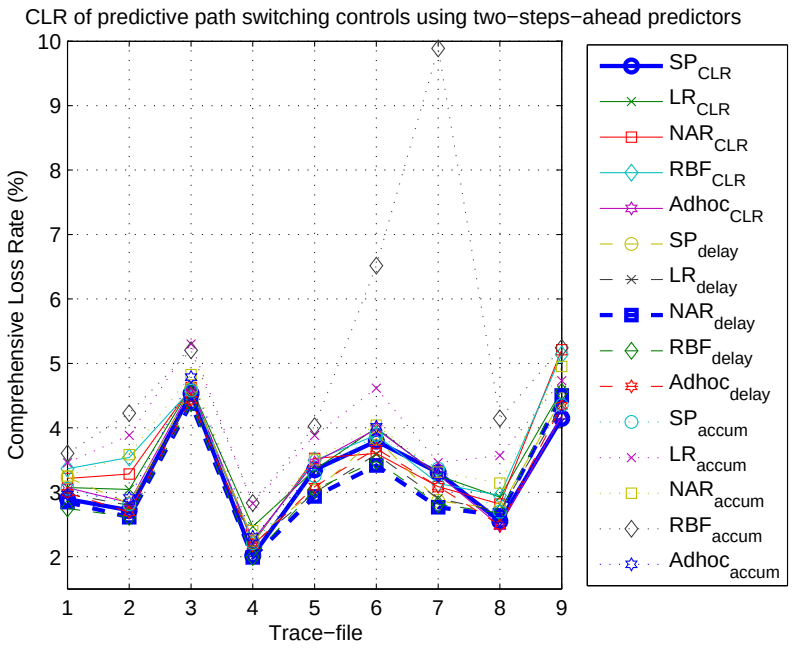

Fig. 52. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A_{new} and Path B_{new} in terms of CLR. SP_{CLR} and the best method, NAR_{delay}, are marked with thicker lines.

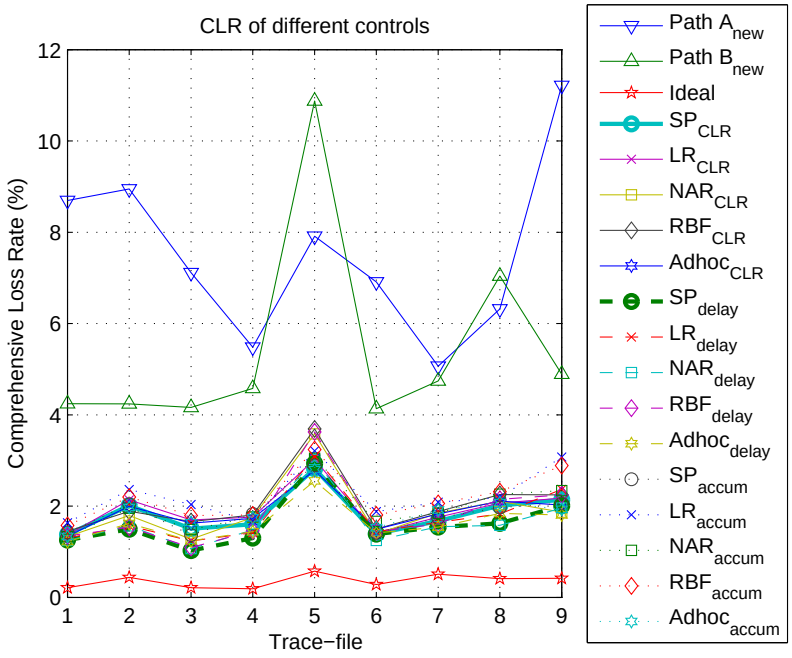

Fig. 53. Results of predictive path switching with different predictors between Path A_{new} and Path C_{new} in terms of CLR. SP_{CLR} and the best method, SP_{delay}, are marked with thicker lines.

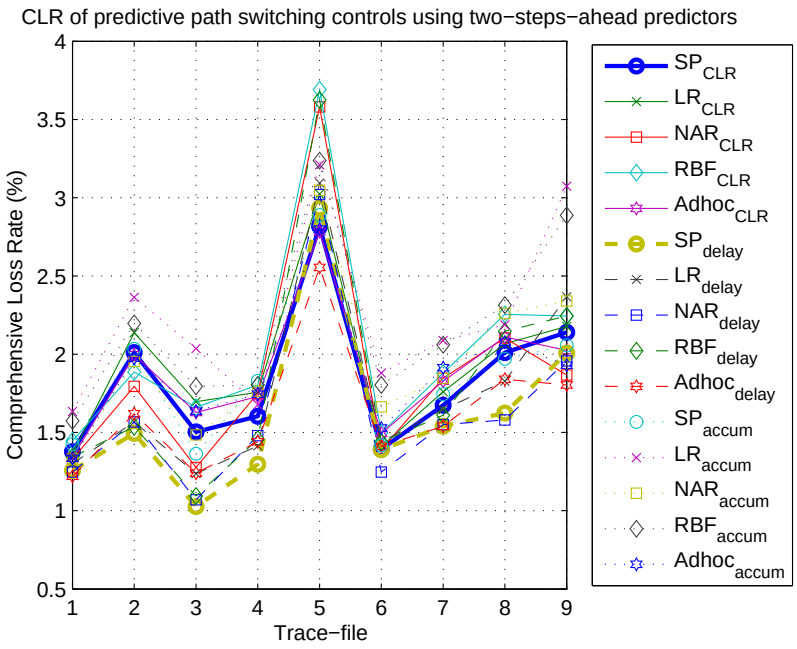

Fig. 54. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A_{new} and Path C_{new} in terms of CLR. SP_{CLR} and the best method, SP_{delay}, are marked with thicker lines.

case path switching control. The NAR_{delay} gives the best results in terms of MOS for predictive path switching control between Path A_{new} and Path B_{new}. The resulting MOS of NAR_{delay} is almost always higher (better) than those of other predictors. The plots for path pair $A_{new}C_{new}$ are in Fig. 57 and Fig. 58. The plots show that the resulting E-model MOS of the predictive path switching controls are higher (better) than those of no switching methods and lower (worse) than those of the ideal case path switching control. The SP_{delay} gives the best results in terms of MOS for predictive path switching control between Path A_{new} and Path C_{new}. The resulting MOS of SP_{delay} is almost always higher (better) than those of other predictors.

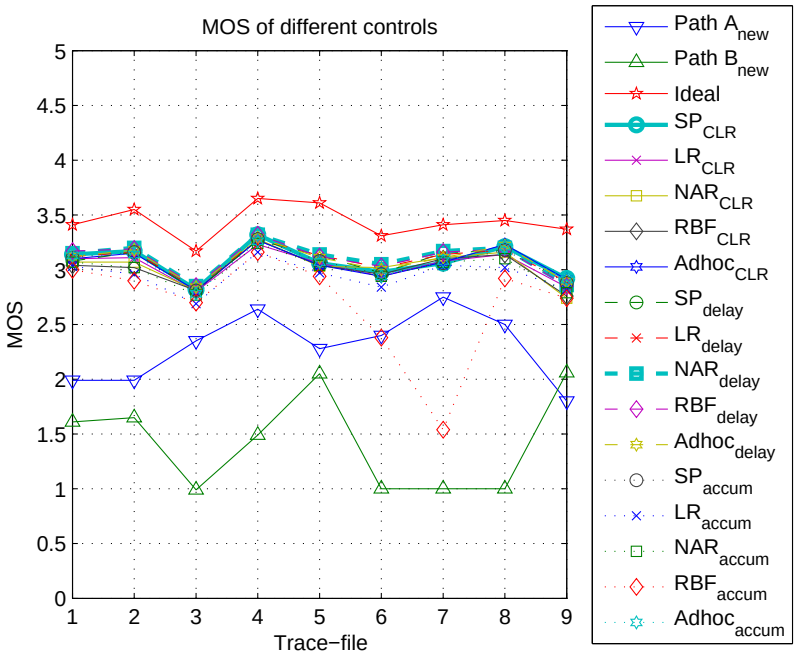

Fig. 55. Results of predictive path switching with different predictors between Path A_{new} and Path B_{new} in terms of E-model MOS. SP_{CLR} and the best method, NAR_{delay}, are marked with thicker lines.

Table XXX. Predictive path switching results between Path A_{new} and Path B_{new} in terms of E-model MOS.

Method	MOS of controls on trace-file									Avg.	The top
	1	2	3	4	5	6	7	8	9		five
Path A_{new}	1.99	1.99	2.35	2.64	2.28	2.40	2.75	2.50	1.80	2.30	
Path B_{new}	1.61	1.65	0.99	1.49	2.05	1.00	1.00	1.00	2.06	1.43	
CLR											
Ideal	3.41	3.55	3.17	3.65	3.61	3.31	3.41	3.45	3.37	3.44	
SP	3.14	3.17	2.82	3.32	3.06	2.98	3.06	3.21	2.92	3.08	5
AR	3.10	3.11	2.80	3.23	3.06	2.94	3.08	3.14	2.84	3.03	
NAR	3.07	3.07	2.81	3.28	3.03	3.02	3.11	3.16	2.74	3.03	
RBF	3.04	3.02	2.81	3.27	3.04	2.96	3.10	3.14	2.76	3.02	
ad hoc	3.10	3.15	2.81	3.27	3.04	2.94	3.07	3.23	2.91	3.06	
Delay											
SP	3.08	3.16	2.81	3.27	3.13	2.97	3.15	3.18	2.87	3.07	
AR	3.12	3.16	2.83	3.28	3.13	3.02	3.14	3.18	2.89	3.08	4
NAR	3.15	3.20	2.85	3.32	3.14	3.05	3.17	3.20	2.86	3.10	1
RBF	3.17	3.19	2.85	3.32	3.12	3.03	3.16	3.19	2.85	3.10	2
ad hoc	3.12	3.18	2.84	3.29	3.11	3.00	3.11	3.22	2.89	3.09	3
Accum.											
SP	3.11	3.16	2.81	3.28	3.08	2.96	3.06	3.18	2.88	3.06	
AR	3.03	2.96	2.69	3.16	2.97	2.84	3.04	3.02	2.82	2.95	
NAR	3.07	3.02	2.77	3.24	3.04	2.94	3.07	3.10	2.79	3.00	
RBF	3.00	2.90	2.70	3.16	2.94	2.38	1.54	2.92	2.74	2.70	
Adhoc	3.13	3.14	2.78	3.26	3.07	2.94	3.08	3.21	2.91	3.06	

Table XXXI. Predictive path switching results between Path A_{new} and Path C_{new} in terms of E-model MOS.

Method	MOS of controls on trace-file									Avg.	The top five
	1	2	3	4	5	6	7	8	9		
Path A_{new}	1.99	1.99	2.35	2.64	2.28	2.40	2.75	2.50	1.80	2.30	
Path C_{new}	2.90	2.90	2.92	2.85	1.89	2.92	2.80	2.41	2.78	2.71	
CLR											
Ideal	3.69	3.65	3.69	3.70	3.62	3.68	3.63	3.65	3.65	3.66	
SP	3.45	3.32	3.42	3.40	3.16	3.44	3.39	3.32	3.29	3.35	
AR	3.45	3.29	3.38	3.37	3.12	3.44	3.37	3.31	3.29	3.34	
NAR	3.46	3.36	3.47	3.37	3.02	3.45	3.35	3.30	3.35	3.35	
RBF	3.43	3.34	3.39	3.36	3.00	3.42	3.35	3.27	3.27	3.32	
Adhoc	3.45	3.32	3.40	3.38	3.17	3.42	3.36	3.30	3.32	3.35	
Delay											
SP	3.47	3.42	3.52	3.46	3.14	3.45	3.41	3.40	3.32	3.40	1
AR	3.46	3.41	3.48	3.44	3.11	3.44	3.39	3.36	3.25	3.37	4
NAR	3.47	3.41	3.51	3.43	3.13	3.47	3.41	3.41	3.33	3.40	3
RBF	3.45	3.41	3.51	3.43	3.02	3.44	3.40	3.29	3.27	3.36	5
Adhoc	3.48	3.40	3.48	3.44	3.21	3.44	3.41	3.35	3.36	3.40	2
Accum.											
SP	3.44	3.31	3.45	3.36	3.15	3.44	3.37	3.33	3.31	3.35	
AR	3.39	3.25	3.31	3.39	3.09	3.35	3.30	3.29	3.12	3.27	
NAR	3.46	3.33	3.42	3.37	3.12	3.39	3.35	3.27	3.26	3.33	
RBF	3.40	3.28	3.36	3.36	3.08	3.36	3.31	3.26	3.15	3.29	
Adhoc	3.45	3.33	3.39	3.37	3.17	3.42	3.34	3.31	3.34	3.35	

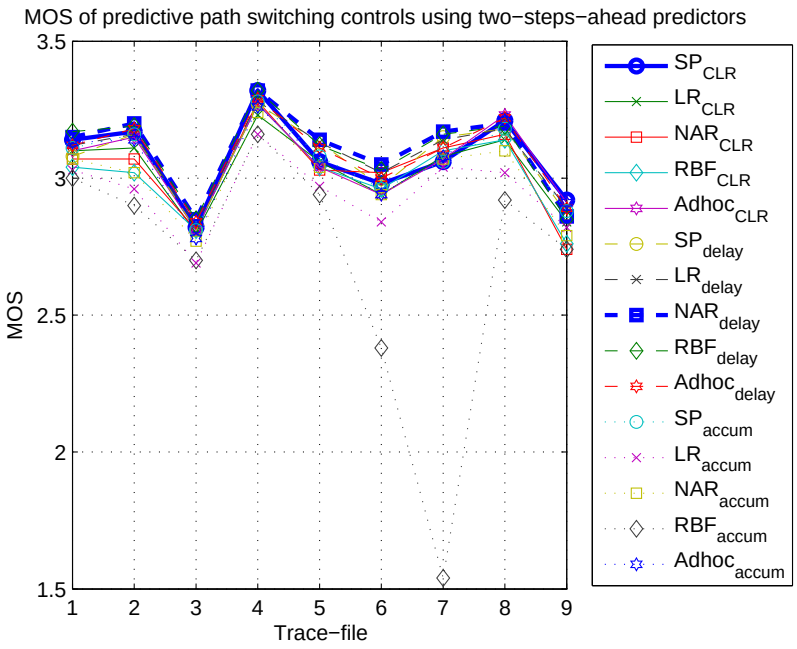

Fig. 56. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A_{new} and Path B_{new} in terms of E-model MOS. SP_{CLR} and the best method, NAR_{delay}, are marked with thicker lines.

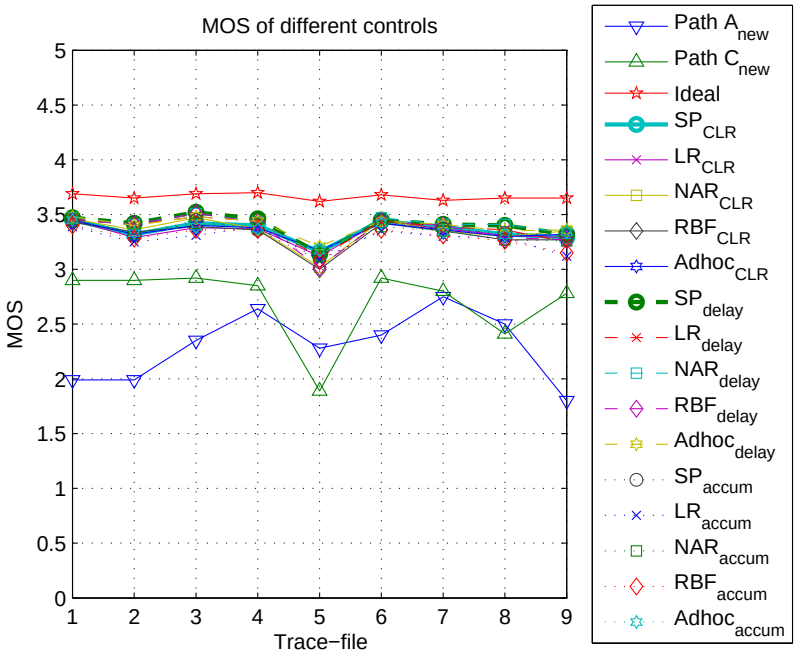

Fig. 57. Results of predictive path switching with different predictors between Path A_{new} and Path C_{new} in terms of E-model MOS. SP_{CLR} and the best method, SP_{delay}, are marked with thicker lines.

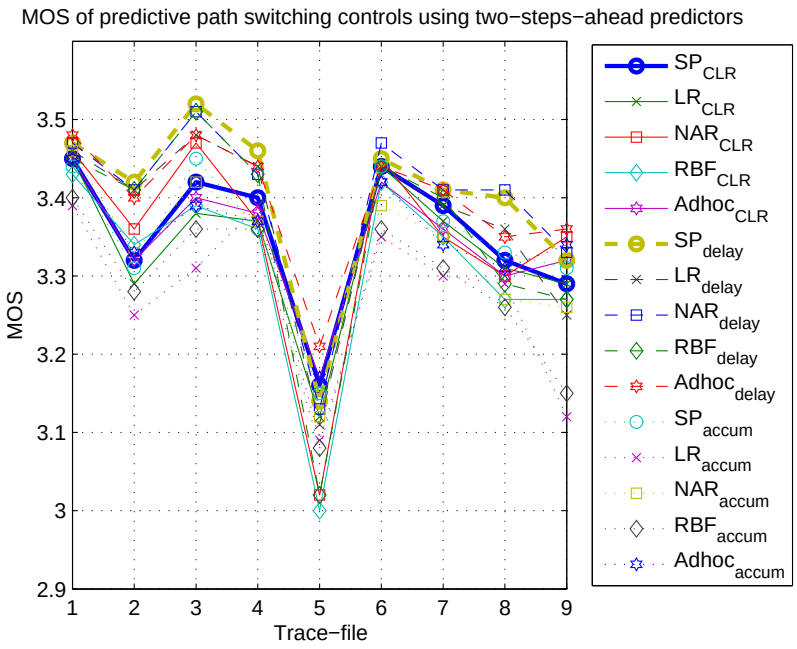

Fig. 58. Zoom-in plot on results of predictive path switching with two-step-ahead predictors between Path A_{new} and Path C_{new} in terms of E-model MOS. SP_{CLR} and the best method, SP_{delay}, are marked with thicker lines.

C. Voting Based Predictive Path Switching Control

From the previous sections (Sec. 1, 2), it is clear that there is no one predictor that is the best for predictive path switching control of all path pairs. In order to have a predictive path switching controller that gives good control results regardless of which path pair is used, a combination of the available predictive path switching controls should be used. In this section a voting scheme is used to combine the decisions of different predictive path switching controllers.

1. The method

The voting scheme is as follows. Seven predictors from the top five predictors of path pair AB, and top five predictors of path pair AC are selected. They are NAR_{CLR}, RBF_{CLR}, $\text{Adhoc}_{\text{CLR}}$, AR_{delay}, $\text{NAR}_{\text{delay}}$, $\text{RBF}_{\text{delay}}$, and $\text{Adhoc}_{\text{delay}}$. Without loss of generality, take switching between Path A and Path B, for example. Each predictor is used to generate one control decision on whether in the next control interval the packets should be transmitted over Path A or over Path B. Each decision is counted as one vote; there are seven votes altogether. The path with the most votes is selected. The pseudocode is as follows:

1. for $i = 1$ to 7
   if the i^{th} method picks Path A
   $D_i = 1$;
   else if the i^{th} method picks Path B
   $D_i = -1$;
   end if
   end for

2. $D_{\text{final}} = \sum_{i=1}^{7} D_i$;

   if $D_{\text{final}} > 0$

   send through Path A;

   else if $D_{\text{final}} < 0$

   send through Path B;

   end

2. Control results

The voting based predictive path switching control results of each path pair are given in Table XXXII. The results for SP_{CLR} and the results of the best predictive path switching control for a given path pair are also given. The results are also plotted in Fig. 59.

The voting based predictive path switching control results for each path pair in terms of E-model MOS are given in Table XXXIII. The E-model MOS results for SP_{CLR} and the best predictive path switching control for a given path pair are also given. The E-model MOS results are also plotted in Fig. 60.

It can be seen that the voting method works well most of the time. Its performance is close to the best predictive path switching control for switching between any pair of paths, either in terms of CLR or in terms of E-model MOS. So the voting based predictive path switching control acts as a universal predictive path switching controller which is independent of the particular path pair under consideration.

3. Section summary

In this section, voting based predictive path switching control, which uses the voted control decision from seven different predictive path switching controllers is proposed. The results show that the voting based predictive path switching control provides a

Table XXXII. Predictive path switching results of the voting based method.

Path Pair	Method	CLR of predictive path switching on trace-file (%)									Avg. (%)
		1	2	3	4	5	6	7	8	9	
AB	SP_{CLR}	9.11	4.48	5.21	6.85	5.83	6.07	6.31	5.56	5.02	6.05
	NAR_{CLR}	8.59	3.36	4.66	5.87	5.01	5.44	5.74	4.84	4.43	5.33
	Voting7[1]	8.36	3.51	4.50	5.69	5.05	5.43	5.49	4.88	4.23	5.24
AC	SP_{CLR}	4.81	2.55	3.52	3.27	3.05	1.82	3.27	3.77	3.25	3.26
	$Adhoc_{delay}$	4.72	2.12	3.01	3.00	2.65	1.64	2.96	3.49	3.21	2.98
	Voting7	4.71	2.06	3.01	2.99	2.45	1.73	2.87	3.54	3.58	3.00
$A_{new}B_{new}$	SP_{CLR}	2.89	2.72	4.53	2.02	3.35	3.79	3.31	2.55	4.15	3.26
	NAR_{delay}	2.85	2.61	4.38	1.99	2.94	3.42	2.77	2.64	4.50	3.12
	Voting7	2.79	2.58	4.36	1.90	2.97	3.39	2.89	2.48	4.46	3.09
$A_{new}C_{new}$	SP_{CLR}	1.38	2.01	1.50	1.60	2.82	1.39	1.67	2.01	2.14	1.84
	SP_{delay}	1.26	1.49	1.03	1.30	2.93	1.39	1.54	1.62	2.01	1.62
	Voting7	1.28	1.57	1.11	1.49	2.62	1.23	1.54	1.78	1.74	1.60

[1] The seven methods used in the voting scheme are:

NAR_{CLR}, RBF_{CLR}, $Adhoc_{CLR}$, AR_{delay}, NAR_{delay}, RBF_{delay}, $Adhoc_{delay}$.

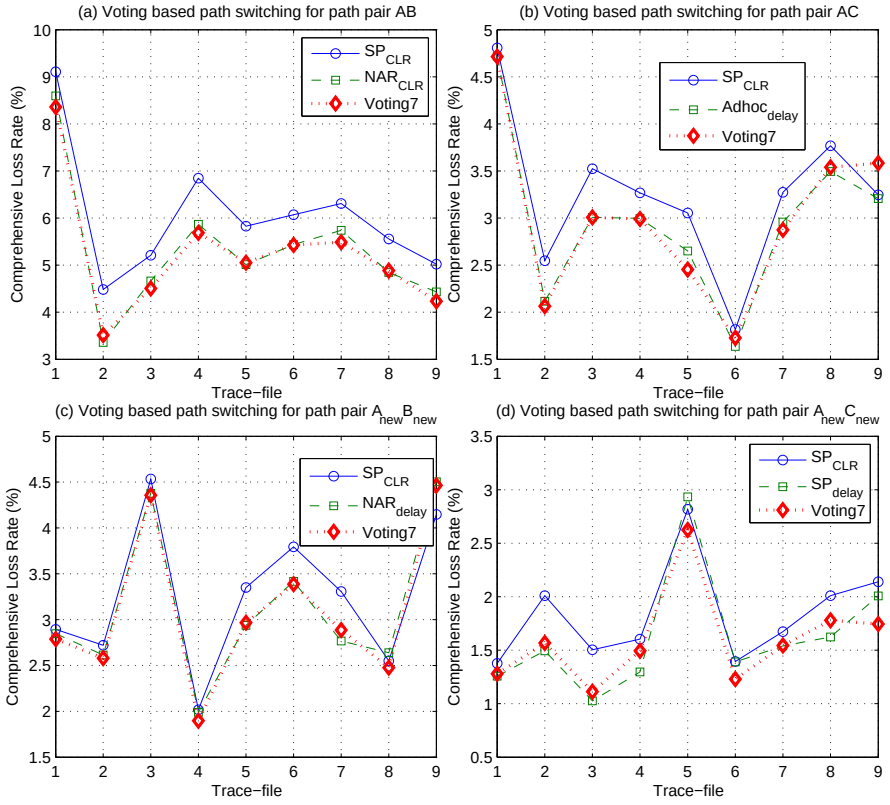

Fig. 59. Voting based predictive path switching control results in terms of CLR. (a) Between path pair AB; (b) Between path pair AC; (c) Between path pair $A_{new}B_{new}$. (d) Between path pair $A_{new}C_{new}$.

Table XXXIII. Predictive path switching results of the voting based method in terms of E-model MOS.

Path Pair	Method	MOS of predictive path switching on trace-file									Avg.
		1	2	3	4	5	6	7	8	9	
AB	SP_{CLR}	2.14	2.86	2.74	2.45	2.63	2.60	2.56	2.69	2.77	2.60
	NAR_{CLR}	2.19	3.06	2.82	2.60	2.76	2.69	2.64	2.79	2.86	2.71
	Voting7	2.23	3.03	2.86	2.63	2.75	2.70	2.69	2.80	2.90	2.73
AC	SP_{CLR}	2.81	3.21	3.03	3.08	3.12	3.36	3.08	2.98	3.08	3.08
	$Adhoc_{delay}$	2.82	3.30	3.13	3.13	3.19	3.40	3.14	3.03	3.09	3.14
	Voting7	2.83	3.31	3.13	3.13	3.23	3.38	3.15	3.03	3.02	3.13
$A_{new}B_{new}$	SP_{CLR}	3.14	3.17	2.82	3.32	3.06	2.98	3.06	3.21	2.92	3.08
	NAR_{delay}	3.15	3.20	2.85	3.32	3.14	3.05	3.17	3.20	2.86	3.10
	Voting7	3.16	3.20	2.86	3.34	3.14	3.05	3.15	3.23	2.87	3.11
$A_{new}C_{new}$	SP_{CLR}	3.45	3.32	3.42	3.40	3.16	3.44	3.39	3.32	3.29	3.35
	SP_{delay}	3.47	3.42	3.52	3.46	3.14	3.45	3.41	3.40	3.32	3.40
	Voting7	3.47	3.41	3.50	3.42	3.20	3.48	3.41	3.37	3.37	3.40

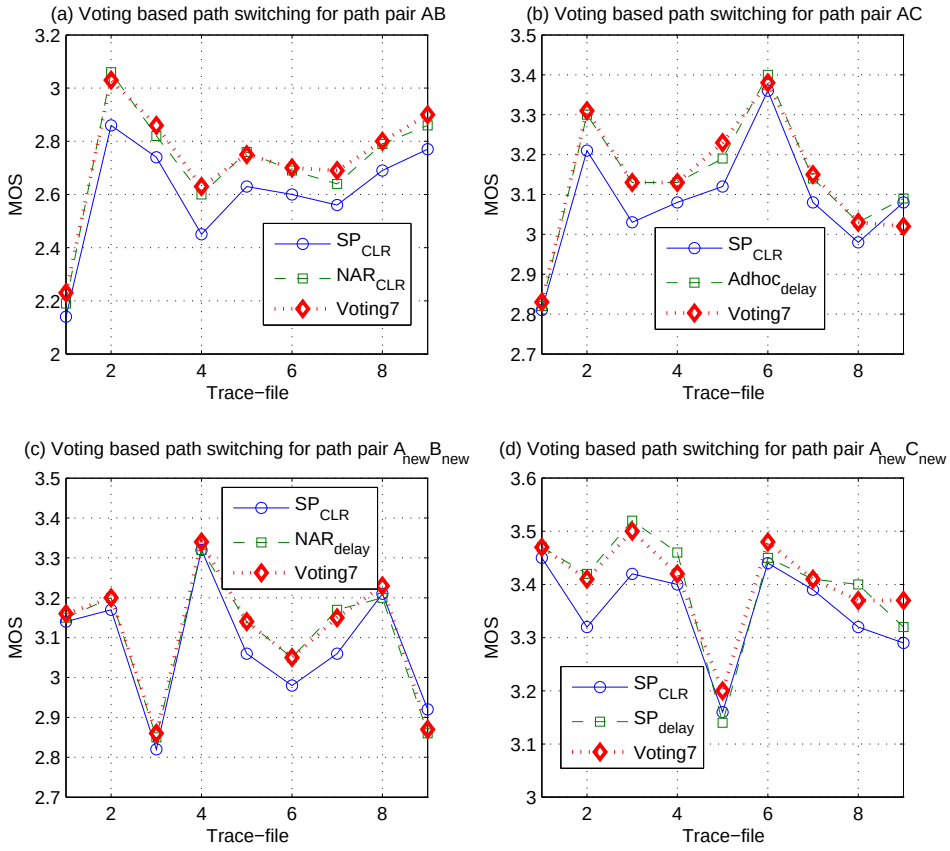

Fig. 60. Voting based predictive path switching control results in terms of E-model MOS. (a) Between path pair AB; (b) Between path pair AC; (c) Between path pair $A_{new}B_{new}$. (d) Between path pair $A_{new}C_{new}$.

universal predictive path switching controller. It provides universally good performance in all cases studied.

D. Study of Predictor Evaluation Criteria

1. Motivation

The previous sections (Sec. 1, 2) reveal that the ranking of the predictors in terms of the MSE prediction criterion used in predictor development does not match their performance ranking on the final results when implemented in predictive path switching control. This means that the MSE prediction criterion does not correlated well with the predictive path switching control performance of the predictors. In this section the reason for the mismatch is investigated and an alternative criterion for the predictor evaluation is discussed.

2. Prediction of the signal difference

One possible reason for the aforementioned mismatch would be that predictors are developed for each path separately, while the path switching control is performed based on the signal difference between a pair of paths. In order to see if this can explain the difference, the difference between the predicted signals of a pair of paths is compared to the real signal difference of that path pair. The resulting MSE of path pair AB and path pair AC are given in Table XXXIV. As the predictors are developed from the first four trace-files of Path A, B, and C, only the average of the last five trace-files are considered for the ranking of the predictors. The resulting MSE of path pair $A_{new}B_{new}$ and $A_{new}C_{new}$ are given in Table XXXV. The comparison of the signal difference prediction results and the predictive path switching control results on the original trace-files and on the new trace-files are listed in Table XXXVI

and XXXVII, respectively.

The results show that the ranking of the predictors for predicting the signal difference is almost the same as their ranking for predicting the individual signals. Except for the RBF predictors for accumulation signals, the signal difference prediction MSE result for accumulation signals is smaller than that of delay signals, and that of delay signals is smaller than CLR signals; and for the prediction of the same kind of information signal, on average, AR, NAR, and RBF predictors are better than ad hoc predictors, and ad hoc predictors are better than SP predictors. But, as shown in Table XXXVI and Table XXXVII, the ranking of the predictors does not match with their ranking of predictive path switching control performance. Even within the same type of information signal the rankings of the prediction results and the control results do not match. This means that whether the predictors are tested for predicting the signal of the individual paths or for the signal difference between two paths does not constitute the reason for the mismatch of the prediction and control rankings. This suggests that the MSE itself might not be a good criterion for judging which predictor is better when it comes to the predictive path switching control.

3. Alternative predictor comparison criterion

Consider the difference between the MSE of the predictions of the signal difference and how the predictions are used in the control loop. Assume that the signal difference is $y_i(k)$ and its prediction is $\hat{y}_i(k|k-d)$, where i stands for the different type of information signals, i.e. *CLR*, *delay*, or *accumulation*, and d is the number of prediction steps ahead. The cost function MSE is calculated by

$$C_{\mathrm{MSE}}(y_i, \hat{y}_i) = \frac{\sum_{k=1}^{N} J_{i,\mathrm{MSE}}(k)}{\sum_{k=1}^{N} y_i^2(k)} \times 100\%, \tag{8.1}$$

Table XXXIV. MSE of the prediction results of the signal difference between paths in the original trace-files.

Path pair	Method	MSE of predicting trace-file (%)									Avg. of
		1	2	3	4	5	6	7	8	9	$5 \sim 9$ (%)
AB	CLR										
	SP	65.57	84.20	73.93	73.48	87.20	89.42	85.61	63.99	57.99	76.84
	AR	51.85	65.05	57.62	58.26	67.63	68.78	66.41	50.99	46.18	60.00
	NAR	51.58	62.86	55.57	57.52	65.44	67.11	64.83	50.85	46.60	58.96
	RBF	49.81	61.90	54.11	56.62	64.88	64.95	63.38	48.03	44.67	57.18
	Adhoc	54.21	69.08	61.18	60.61	71.38	73.20	70.31	52.71	47.86	63.09
	Delay										
	SP	34.51	50.58	36.77	50.46	56.88	49.96	50.81	30.16	31.52	43.87
	AR	30.05	43.29	32.27	43.02	49.23	43.40	44.80	26.30	27.39	38.23
	NAR	29.85	41.94	31.29	42.22	47.47	42.18	43.22	25.94	27.28	37.22
	RBF	29.26	39.72	29.59	41.31	45.50	40.51	42.04	24.93	27.13	36.02
	Adhoc	32.51	46.92	37.59	46.22	52.07	47.59	48.37	28.64	29.86	41.30
	Accum.										
	SP	14.31	36.80	24.26	28.67	44.40	17.36	33.68	13.57	17.13	25.23
	AR	13.44	33.32	22.28	26.08	40.41	16.60	31.43	12.63	15.59	23.33
	NAR	15.35	31.38	21.41	25.41	38.16	17.39	30.16	13.15	15.89	22.95
	RBF	80.29	30.46	20.39	34.14	38.40	36.13	31.89	23.17	25.74	31.07
	Adhoc	13.80	35.05	23.28	27.46	42.21	16.90	32.21	13.03	16.40	24.15
AC	CLR										
	SP	41.24	81.69	81.94	77.76	71.11	68.32	63.25	42.28	51.39	59.27
	AR	35.16	65.72	65.31	61.75	57.83	54.73	51.52	36.23	42.18	48.50
	NAR	41.32	64.40	64.21	61.69	58.51	54.49	54.03	38.07	45.58	50.14
	RBF	33.75	63.09	62.55	60.33	56.74	53.50	51.09	34.87	42.55	47.75
	Adhoc	34.28	66.86	67.12	63.75	58.62	56.19	52.70	36.01	41.97	49.10
	Delay										
	SP	21.07	54.26	55.91	49.66	47.49	39.62	34.76	21.65	27.90	34.28
	AR	18.47	46.54	47.92	43.03	40.44	34.74	31.92	19.38	24.51	30.20
	NAR	18.91	45.52	46.76	42.18	39.93	34.74	31.27	19.48	24.36	29.96
	RBF	22.38	40.35	43.32	40.13	40.95	32.94	33.21	19.09	25.93	30.42
	Adhoc	18.79	47.88	49.22	44.22	42.14	35.72	31.85	19.92	24.70	30.86
	Accum.										
	SP	9.13	44.79	35.46	31.95	41.39	13.49	21.12	10.35	13.84	20.04
	AR	8.44	40.22	32.56	29.22	36.17	13.03	20.07	9.62	12.84	18.35
	NAR	11.68	38.02	38.74	28.95	34.54	14.22	21.16	10.65	16.39	19.39
	RBF	26.27	35.89	37.34	31.31	38.88	34.45	29.69	21.61	24.68	29.86
	Adhoc	9.08	44.50	35.23	31.75	41.12	13.43	21.00	10.29	13.76	19.92

Table XXXV. MSE of the prediction results of the signal difference between paths in the new trace-files.

Path pair	Method	MSE of predicting trace-file (%)									Avg.
		1	2	3	4	5	6	7	8	9	(%)
$A_{new}B_{new}$	CLR										
	SP	49.45	59.50	42.84	54.34	82.47	44.33	28.23	34.69	70.60	51.83
	AR	41.02	48.95	35.85	44.62	65.49	37.35	25.88	29.74	56.82	42.86
	NAR	43.87	50.75	35.80	44.45	65.94	37.75	25.78	30.19	59.29	43.76
	RBF	43.62	50.89	35.89	44.72	66.63	37.82	25.59	30.35	59.37	43.87
	Adhoc	41.67	49.87	36.45	46.79	68.21	37.90	25.82	29.64	59.28	43.96
	Delay										
	SP	29.19	43.68	22.83	34.58	57.95	30.23	18.52	19.69	40.02	32.96
	AR	26.23	38.41	21.06	30.67	51.79	27.33	17.25	18.08	36.94	29.75
	NAR	26.83	36.78	20.93	29.56	49.61	27.44	17.80	18.03	36.08	29.23
	RBF	24.70	35.87	20.65	28.81	47.79	27.43	18.11	17.98	35.50	28.54
	Adhoc	29.12	40.83	24.82	36.71	52.59	30.53	22.60	21.30	41.35	33.32
	Accum.										
	SP	16.53	25.81	18.10	20.82	26.48	12.62	6.10	9.74	24.91	17.90
	AR	16.08	24.89	16.64	19.97	25.15	11.85	6.03	9.23	23.25	17.01
	NAR	16.26	24.94	18.28	20.08	28.28	17.06	12.72	11.67	22.66	19.11
	RBF	25.55	26.91	17.47	24.41	94.31	100.38	143.37	40.52	24.24	55.24
	Adhoc	15.92	24.70	17.31	20.09	25.45	12.33	6.14	9.45	23.98	17.26
$A_{new}C_{new}$	CLR										
	SP	37.51	52.44	55.07	61.53	69.32	56.70	64.38	56.48	62.51	57.33
	AR	33.28	45.45	47.76	52.73	56.81	48.45	54.24	48.69	52.02	48.82
	NAR	34.63	45.17	47.86	52.27	57.76	48.39	53.52	49.33	52.16	49.01
	RBF	32.04	43.76	46.51	51.69	58.27	47.25	52.84	47.83	50.97	47.91
	Adhoc	32.83	45.65	48.35	53.32	57.11	49.73	55.48	49.39	53.07	49.44
	Delay										
	SP	21.04	32.54	36.54	38.05	47.70	38.52	42.71	38.02	38.47	37.07
	AR	18.79	29.11	32.27	34.10	42.75	34.28	38.37	34.08	35.07	33.20
	NAR	20.48	27.97	30.80	31.78	41.31	31.93	36.81	33.89	33.88	32.10
	RBF	19.48	25.66	28.72	29.48	43.30	30.07	36.58	33.93	32.30	31.06
	Adhoc	19.74	30.50	34.14	35.55	42.41	35.96	39.27	35.28	35.52	34.27
	Accum.										
	SP	12.02	17.51	30.56	27.60	42.10	30.70	37.44	31.67	26.32	28.43
	AR	11.31	16.54	28.08	25.24	37.54	28.22	34.01	29.11	24.50	26.06
	NAR	11.81	17.34	27.55	25.38	36.33	27.58	32.49	30.31	23.81	25.84
	RBF	25.55	26.91	17.47	24.41	94.31	100.38	143.37	40.52	24.24	55.24
	Adhoc	11.95	17.41	30.37	27.43	41.82	30.51	37.20	31.47	26.17	28.26

Table XXXVI. Comparison of the signal difference prediction results and the predictive path switching control results on the original trace-files.

Information signal	Method	Avg. of trace-files ($5 \sim 9$)			Ranking					
					Same signal[1]			Over all[2]		
		MSE (%)	CLR(%)	MOS	MSE	CLR	MOS	MSE	CLR	MOS
Path Pair AB										
CLR	SP	76.84	5.76	2.65	5	5	5	15	10	10
	AR	60.00	5.60	2.67	3	4	4	13	7	7
	NAR	58.96	5.09	2.75	2	1	1	12	1	1
	RBF	57.18	5.10	2.75	1	2	2	11	2	2
	Adhoc	63.09	5.39	2.70	4	3	3	14	5	5
Delay	SP	43.87	6.92	2.48	5	5	5	10	15	15
	AR	38.23	5.94	2.63	3	4	4	8	12	12
	NAR	37.22	5.22	2.74	2	1	1	7	3	3
	RBF	36.02	5.23	2.74	1	2	2	6	4	4
	Adhoc	41.30	5.82	2.64	4	3	3	9	11	11
Accum.	SP	25.23	5.99	2.62	4	4	4	4	13	13
	AR	23.33	6.09	2.59	2	5	5	2	14	14
	NAR	22.95	5.48	2.69	1	1	1	1	6	6
	RBF	31.07	5.69	2.65	5	3	3	5	9	9
	Adhoc	24.15	5.62	2.67	3	2	2	3	8	8
Path Pair AC										
CLR	SP	59.27	3.03	3.12	5	2	2	15	6	6
	AR	48.50	3.16	3.10	2	3	3	12	10	10
	NAR	50.14	3.44	3.05	4	5	5	14	13	13
	RBF	47.75	3.44	3.05	1	4	4	11	12	12
	Adhoc	49.10	3.00	3.13	3	1	1	13	4	4
Delay	SP	34.28	3.01	3.13	5	4	4	10	5	5
	AR	30.20	2.88	3.15	2	2	2	7	2	2
	NAR	29.96	2.95	3.14	1	3	3	6	3	3
	RBF	30.42	3.14	3.10	3	5	5	8	8	8
	Adhoc	30.86	2.79	3.17	4	1	1	9	1	1
Accum.	SP	20.04	3.16	3.10	4	2	2	4	9	9
	AR	18.35	3.48	3.04	1	4	4	1	14	14
	NAR	19.39	3.29	3.08	2	3	3	2	11	11
	RBF	29.86	3.68	3.00	5	5	5	5	15	15
	Adhoc	19.92	3.08	3.11	3	1	1	3	7	7

[1] Rankings for the predictions and for the control results based on the predictors for the same signal.

[2] Rankings for the predictions and for the control results based on all the predictors of that path pair.

Table XXXVII. Comparison of the signal difference prediction results and the predictive path switching control results on the new trace-files.

Information signal	Method	Avg. of trace-files ($1 \sim 9$)			Ranking					
					Same signal			Over all		
		MSE (%)	CLR(%)	MOS	MSE	CLR	MOS	MSE	CLR	MOS
Path pair $A_{new}B_{new}$										
CLR	SP	51.83	3.26	3.08	5	1	1	14	5	5
	AR	42.86	3.48	3.03	1	3	3	10	10	10
	NAR	43.76	3.51	3.03	2	4	4	11	11	11
	RBF	43.87	3.59	3.02	3	5	5	12	12	12
	Adhoc	43.96	3.35	3.06	4	2	2	13	7	7
Delay	SP	32.96	3.32	3.07	4	5	5	8	6	6
	AR	29.75	3.22	3.08	3	3	4	7	3	4
	NAR	29.23	3.12	3.10	2	1	1	6	1	1
	RBF	28.54	3.15	3.10	1	2	2	5	2	2
	Adhoc	33.32	3.22	3.09	5	4	3	9	4	3
Accum.	SP	17.90	3.36	3.06	3	1	1	3	8	8
	AR	17.01	3.97	2.95	1	4	4	1	14	14
	NAR	19.11	3.67	3.00	4	3	3	4	13	13
	RBF	55.24	5.08	2.70	5	5	5	15	15	15
	Adhoc	17.26	3.37	3.06	2	2	2	2	9	9
Path pair $A_{new}C_{new}$										
CLR	SP	57.33	1.84	3.35	5	1	1	15	6	6
	AR	48.82	1.93	3.34	2	4	4	11	11	11
	NAR	49.01	1.88	3.35	3	3	3	12	10	10
	RBF	47.91	2.04	3.32	1	5	5	10	13	13
	Adhoc	49.44	1.88	3.35	4	2	2	13	8	8
Delay	SP	37.07	1.62	3.40	5	1	1	9	1	1
	AR	33.20	1.77	3.37	3	4	4	7	4	4
	NAR	32.10	1.64	3.40	2	3	3	6	3	3
	RBF	31.06	1.83	3.36	1	5	5	5	5	5
	Adhoc	34.27	1.63	3.40	4	2	2	8	2	2
Accum.	SP	28.43	1.86	3.35	4	1	1	4	7	7
	AR	26.06	2.24	3.27	2	5	5	2	15	15
	NAR	25.84	1.97	3.33	1	3	3	1	12	12
	RBF	55.24	2.19	3.29	5	4	4	14	14	14
	Adhoc	28.26	1.88	3.35	3	2	2	3	9	9

where,

$$J_{i,\mathrm{MSE}}(k) \quad = \quad [y_i(k) - \hat{y}_i(k|k-d)]^2. \tag{8.2}$$

The penalty of the prediction error in each term is quadratic. The prediction is penalized if it is away from the real signal on either side, as shown in Fig. 61. In Fig. 61, the point $y_i(k)$ is the real value and the curve $J_{i,\mathrm{MSE}}(k)$ shows that the prediction $\hat{y}_i(k|k-d)$ is quadratically penalized when it is larger or smaller than $y_i(k)$ regardless whether it has the same sign as $y_i(k)$. The further away the prediction $\hat{y}_i(k|k-d)$ is from the true value $y_i(k)$ the higher the penalty is.

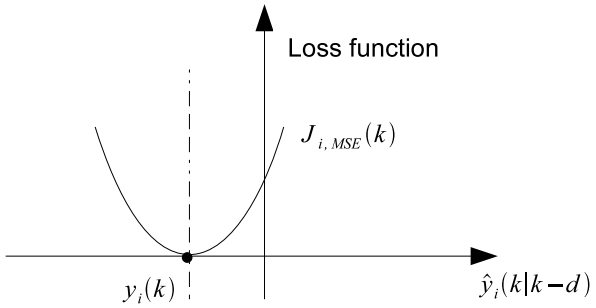

Fig. 61. Plot of the quadratic prediction error criterion.

But in the predictive path switching control, what really matters is the sign of the signal difference. In the case of the available paths, if it is negative the packets will be transmitted over the first path, if it is positive then packets will be transmitted over the second path. So the penalty of prediction for each term should be a zero-one function, which has the shape as shown in Fig. 62, depending on the sign of the predicted signal difference and that of the true CLR signal.

If the predicted signal difference $\hat{y}_i(k|k-d)$ has the same sign as the real CLR signal difference $y_{CLR}(k)$, the prediction should not be penalized. If the predicted

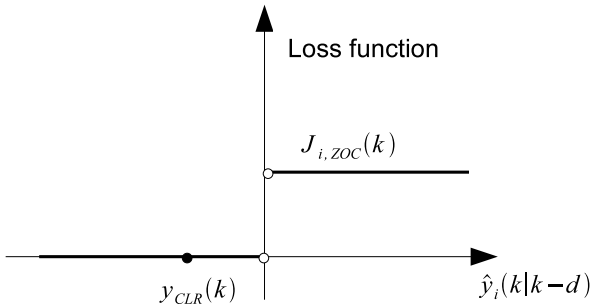

Fig. 62. Plot of the zero-one prediction performance criterion.

signal difference $\hat{y}_i(k|k-d)$ has the wrong sign compared to the CLR signal $y_{CLR}(k)$, then the prediction should be penalized with a constant cost. If the real CLR signal difference is zero, then it doesn't matter what the prediction is, no matter over which path the packets are transmitted, the results will be the same, because at that time the two paths have same quality. If the the real CLR signal difference $y_{CLR}(k)$ is not zero, but the prediction is zero, then the latest nonzero prediction result is used for estimating the prediction quality, because that prediction value is used in the control.

The proposed new Zero-One error Criterion (ZOC) is calculated in two steps:

1. Modify the predicted signal differences to match the predictions used in control.

$$\tilde{\hat{y}}_i(k|k-d) = \begin{cases} \hat{y}_i(k|k-d), & \hat{y}_i(k|k-d) \neq 0; \\ \epsilon \cdot \mathrm{sgn}(\hat{y}_i(k-l)|k-d-l), & \hat{y}_i(k-j|k-d-j) = 0, \\ & \hat{y}_i(k-l|k-d-l) \neq 0, 0 \leq j < l. \end{cases} , \tag{8.3}$$

   where $0 < \epsilon \ll 1$.

2. The penalty of the predicted signal difference of each term is given by the ZOC

cost function

$$J_{i,\mathrm{ZOC}}(k) = \begin{cases} 0 & \mathrm{sgn}(y_i(k))\mathrm{sgn}(\tilde{\hat{y}}_{CLR}(k|k-d)) \geq 0; \\ 1 & \mathrm{sgn}(y_i(k))\mathrm{sgn}(\tilde{\hat{y}}_{CLR}(k|k-d)) < 0. \end{cases} \tag{8.4}$$

The overall cost function is

$$C_{\mathrm{ZOC}}(y_{CLR}, \hat{y}_i) = \frac{1}{N}\sum_{k=1}^{N} J_{i,\mathrm{ZOC}}(k) \times 100\%. \tag{8.5}$$

The results of using this ZOC are given in Table XXXVIII and Table XXXIX. Table XL and Table XLI shows the comparison of the ranking of the prediction results in the ZOC sense with that of predictive path switching control results in the CLR sense.

The results show that the ranking of the prediction results in the ZOC sense is far close to the ranking of the predictive path switching control results in the CLR sense. The proposed ZOC is more correlated with the predictive path switching control results than the original MSE criterion. But when the prediction results are close to each other, with ZOC within 0.1% of each other, there are still cases of mismatch in the ranking of the prediction results in terms of the ZOC compared to that of the predictive path switching control results. This is because in this criterion only the signal difference is taken into account, while the real CLR at that time interval is not taken into account. This makes a difference in the predictive path switching control results, especially when the ZOCs are close to each other. In this ZOC, all the sign errors are treated in the same manner, but for the control, a prediction error in a time interval where the CLR is low is different from a prediction error in a time interval where the CLR is high. Also, improving the development of predictors using this criterion is still an open question. As the ranking of the predictive path switching control results in the MOS sense is the same as the ranking of the predictive path

Table XXXVIII. ZOC of the prediction results of the signal difference between paths in the original trace-files.

Path pair	Information signal	Method	ZOC of predicting trace-file (%)									Avg. of $5 \sim 9$ (%)
			1	2	3	4	5	6	7	8	9	
AB	CLR	SP	7.74	5.28	6.20	7.35	7.06	7.03	6.99	4.28	4.55	5.98
		AR	7.33	4.84	6.08	6.37	6.74	6.84	6.74	4.24	4.19	5.75
		NAR	6.95	3.84	5.66	5.77	5.88	6.33	6.24	3.37	3.77	5.12
		RBF	6.97	3.91	5.55	5.84	6.01	6.19	6.14	3.42	3.66	5.08
		Adhoc	7.51	4.39	6.00	6.22	6.22	6.30	6.85	3.82	4.07	5.45
	Delay	SP	8.46	6.84	6.69	8.45	8.53	7.83	8.64	6.65	7.00	7.73
		AR	7.47	5.74	6.03	6.70	7.12	7.45	6.81	5.05	4.97	6.28
		NAR	7.04	4.20	5.53	6.26	6.42	6.28	6.08	3.61	4.05	5.29
		RBF	6.87	4.27	5.28	6.34	6.37	6.05	6.05	3.61	4.31	5.28
		Adhoc	7.99	5.85	6.39	6.70	6.87	7.15	7.01	4.35	4.60	6.00
	Accum.	SP	8.88	5.78	6.27	7.46	7.72	6.84	6.91	4.50	5.15	6.22
		AR	8.08	5.68	6.12	7.76	7.49	6.85	7.43	4.84	5.62	6.45
		NAR	7.66	4.72	5.74	6.64	6.81	6.34	6.37	3.99	4.47	5.60
		RBF	8.19	4.87	6.08	7.08	7.03	6.47	6.49	4.24	5.08	5.86
		Adhoc	7.95	4.96	6.16	6.61	6.69	6.28	6.69	4.12	4.60	5.68
AC	CLR	SP	4.12	3.84	4.88	4.18	4.35	2.69	3.96	2.88	3.80	3.54
		AR	4.62	3.68	5.19	4.38	4.53	2.93	4.07	2.76	4.34	3.73
		NAR	4.53	3.68	4.72	3.91	4.26	2.76	4.07	2.99	6.42	4.10
		RBF	4.64	3.47	4.77	4.04	4.60	3.72	4.23	3.04	5.30	4.18
		Adhoc	4.81	3.82	4.72	4.26	4.38	2.85	3.85	2.65	3.72	3.49
	Delay	SP	3.92	2.96	4.10	3.65	3.95	2.54	3.49	2.49	5.03	3.50
		AR	4.22	3.28	4.28	4.12	3.72	2.81	3.95	2.35	4.00	3.37
		NAR	4.18	3.11	4.37	3.73	3.80	3.05	3.70	2.51	4.24	3.46
		RBF	4.01	3.11	4.19	3.85	4.07	3.07	3.68	2.49	5.42	3.74
		Adhoc	3.96	3.26	4.08	3.73	3.82	2.34	3.61	2.42	3.73	3.18
	Accum.	SP	4.45	3.62	4.41	4.70	4.55	3.37	3.70	2.99	3.89	3.70
		AR	5.33	4.15	5.39	5.28	5.37	4.19	4.88	3.03	4.11	4.31
		NAR	4.60	3.93	5.00	4.72	4.50	3.45	4.38	2.76	4.47	3.91
		RBF	5.68	4.04	5.45	4.85	5.37	4.30	4.85	2.99	5.07	4.51
		Adhoc	4.74	3.82	4.69	4.04	4.66	2.97	4.04	2.64	3.62	3.59

Table XXXIX. ZOC of the prediction results of the signal difference between paths in the new trace-files.

Path pair	Information signal	Method	ZOC of predicting trace-file (%)									Avg. (%)
			1	2	3	4	5	6	7	8	9	
$A_{new}B_{new}$	CLR	SP	2.41	3.16	3.03	3.03	4.78	2.88	2.87	2.09	4.15	3.16
		AR	2.72	3.82	3.41	3.89	4.85	3.34	2.88	2.97	5.00	3.65
		NAR	3.27	4.27	3.27	3.38	4.91	2.77	2.74	2.54	5.69	3.65
		RBF	3.50	4.66	3.19	3.45	4.82	3.16	2.82	2.74	5.62	3.78
		Adhoc	2.81	3.35	3.24	3.69	5.14	3.20	2.80	2.12	4.35	3.41
	Delay	SP	3.00	3.46	3.50	3.61	4.31	3.15	2.24	2.59	4.81	3.41
		AR	2.58	3.46	3.14	3.51	4.41	2.74	2.34	2.59	4.72	3.28
		NAR	2.51	3.11	2.89	3.07	4.12	2.43	2.18	2.35	4.81	3.05
		RBF	2.28	3.20	2.88	2.96	4.35	2.50	2.18	2.47	4.91	3.08
		ad hoc	2.74	3.26	2.95	3.51	4.60	2.73	2.53	2.12	4.62	3.23
	Accum.	SP	2.73	3.12	3.05	3.32	4.51	3.07	3.03	2.32	4.60	3.31
		AR	3.50	5.26	4.24	4.60	5.81	4.07	3.14	4.18	5.23	4.45
		NAR	3.23	4.69	3.49	3.80	4.96	3.24	2.96	3.15	5.37	3.88
		RBF	3.77	5.87	4.10	4.50	5.74	6.14	9.74	4.69	5.80	5.59
		ad hoc	2.68	3.42	3.49	3.76	4.80	3.18	2.77	2.19	4.22	3.39
$A_{new}C_{new}$	CLR	SP	2.15	2.69	2.11	2.41	3.65	2.01	2.04	2.46	2.85	2.49
		AR	2.20	2.87	2.55	2.61	3.96	2.14	2.15	2.46	2.82	2.64
		NAR	2.01	2.30	1.74	2.49	4.92	2.00	2.22	2.50	2.35	2.50
		RBF	2.27	2.51	2.42	2.81	4.92	2.26	2.30	2.78	2.95	2.80
		Adhoc	2.24	2.66	2.47	2.70	3.58	2.15	2.26	2.54	2.66	2.59
	Delay	SP	1.93	1.89	1.51	2.05	3.76	2.22	1.87	2.19	2.84	2.25
		AR	2.03	2.03	1.89	2.26	4.16	2.26	1.96	2.32	3.27	2.46
		NAR	1.92	1.96	1.55	2.19	3.91	1.89	1.84	1.93	2.54	2.19
		RBF	2.07	2.01	1.59	2.28	4.89	2.24	1.97	2.82	3.03	2.55
		Adhoc	1.93	2.04	1.88	2.23	3.28	2.09	1.85	2.31	2.38	2.22
	Accum.	SP	2.24	2.64	1.99	2.81	3.70	2.08	2.18	2.37	2.76	2.53
		AR	2.80	3.45	3.15	2.81	4.23	2.80	2.74	2.85	4.34	3.24
		NAR	2.15	2.50	2.07	2.68	4.03	2.41	2.22	2.76	2.99	2.64
		RBF	2.54	3.04	2.70	2.82	4.28	2.78	2.65	2.97	3.97	3.09
		Adhoc	2.15	2.61	2.46	2.66	3.64	2.22	2.30	2.45	2.57	2.56

Table XL. Comparison of the prediction results in terms of ZOC and the predictive path switching control results from the original trace-files.

Information signal	Method	Avg. of trace-files ($5 \sim 9$)			Ranking					
					Same signal			Over all		
		ZOC (%)	CLR(%)	MOS	ZOC	CLR	MOS	ZOC	CLR	MOS
Path pair AB										
CLR	SP	5.98	5.76	2.65	5	5	5	10	10	10
	AR	5.75	5.60	2.67	4	4	4	8	7	7
	NAR	5.12	5.09	2.75	2	1	1	2	1	1
	RBF	5.08	5.10	2.75	1	2	2	1	2	2
	Adhoc	5.45	5.39	2.70	3	3	3	5	5	5
Delay	SP	7.73	6.92	2.48	5	5	5	15	15	15
	AR	6.28	5.94	2.63	4	4	4	13	12	12
	NAR	5.29	5.22	2.74	2	1	1	4	3	3
	RBF	5.28	5.23	2.74	1	2	2	3	4	4
	Adhoc	6.00	5.82	2.64	3	3	3	11	11	11
Accum.	SP	6.22	5.99	2.62	4	4	4	12	13	13
	AR	6.45	6.09	2.59	5	5	5	14	14	14
	NAR	5.60	5.48	2.69	1	1	1	6	6	6
	RBF	5.86	5.69	2.65	3	3	3	9	9	9
	Adhoc	5.68	5.62	2.67	2	2	2	7	8	8
Path pair AC										
CLR	SP	3.54	3.03	3.12	2	2	2	6	6	6
	AR	3.73	3.16	3.10	3	3	3	9	10	10
	NAR	4.10	3.44	3.05	4	5	5	12	13	13
	RBF	4.18	3.44	3.05	5	4	4	13	12	12
	Adhoc	3.49	3.00	3.13	1	1	1	4	4	4
Delay	SP	3.50	3.01	3.13	4	4	4	5	5	5
	AR	3.37	2.88	3.15	2	2	2	2	2	2
	NAR	3.46	2.95	3.14	3	3	3	3	3	3
	RBF	3.74	3.14	3.10	5	5	5	10	8	8
	Adhoc	3.18	2.79	3.17	1	1	1	1	1	1
Accum.	SP	3.70	3.16	3.10	2	2	2	8	9	9
	AR	4.31	3.48	3.04	4	4	4	14	14	14
	NAR	3.91	3.29	3.08	3	3	3	11	11	11
	RBF	4.51	3.68	3.00	5	5	5	15	15	15
	Adhoc	3.59	3.08	3.11	1	1	1	7	7	7

Table XLI. Comparison of the prediction results in terms of ZOC and the predictive path switching control results from the new trace-files.

Information signal	Method	Avg. of trace-files ($1 \sim 9$)			Ranking					
					Same signal			Over all		
		ZOC (%)	CLR(%)	MOS	ZOC	CLR	MOS	ZOC	CLR	MOS
Path pair $A_{new}B_{new}$										
CLR	SP	3.16	3.26	3.08	1	1	1	3	5	5
	AR	3.65	3.48	3.03	4	3	3	11	10	10
	NAR	3.65	3.51	3.03	3	4	4	10	11	11
	RBF	3.78	3.59	3.02	5	5	5	12	12	12
	Adhoc	3.41	3.35	3.06	2	2	2	9	7	7
Delay	SP	3.41	3.32	3.07	5	5	5	8	6	6
	AR	3.28	3.22	3.08	4	4	4	5	4	4
	NAR	3.05	3.12	3.10	1	1	1	1	1	1
	RBF	3.08	3.15	3.10	2	2	2	2	2	2
	Adhoc	3.23	3.22	3.09	3	3	3	4	3	3
Accum.	SP	3.31	3.36	3.06	1	1	1	6	8	8
	AR	4.45	3.97	2.95	4	4	4	14	14	14
	NAR	3.88	3.67	3.00	3	3	3	13	13	13
	RBF	5.59	5.08	2.70	5	5	5	15	15	15
	Adhoc	3.39	3.37	3.06	2	2	2	7	9	9
Path pair $A_{new}C_{new}$										
CLR	SP	2.49	1.84	3.35	1	1	1	5	6	6
	AR	2.64	1.93	3.34	4	4	4	11	11	11
	NAR	2.50	1.88	3.35	2	3	3	6	10	10
	RBF	2.80	2.04	3.32	5	5	5	13	13	13
	Adhoc	2.59	1.88	3.35	3	2	2	10	8	8
Delay	SP	2.25	1.62	3.40	3	1	1	3	1	1
	AR	2.46	1.77	3.37	4	4	4	4	4	4
	NAR	2.19	1.64	3.40	1	3	3	1	3	3
	RBF	2.55	1.83	3.36	5	5	5	8	5	5
	Adhoc	2.22	1.63	3.40	2	2	2	2	2	2
Accum.	SP	2.53	1.86	3.35	1	1	1	7	7	7
	AR	3.24	2.24	3.27	5	5	5	15	15	15
	NAR	2.64	1.97	3.33	3	3	3	12	12	12
	RBF	3.09	2.19	3.29	4	4	4	14	14	14
	Adhoc	2.56	1.88	3.35	2	2	2	9	9	9

switching control results in the CLR sense in these results, the ZOC ranking is also close to the MOS ranking.

E. Chapter Summary and Conclusions

In this chapter predictive path switching controllers based on different types of predictors are studied. A voting based predictive path switching control is proposed and investigated. The mismatch between MSE and the predictive path switching control results is discussed. An alternative criterion for predictor evaluation is discussed. From the predictive path switching results between path pairs AB, AC, $A_{new}B_{new}$, and $A_{new}C_{new}$, the following conclusions are obtained:

1. The predictive path switching control is generally better than no switching.
2. There is no one particular type of predictor that is gives the best predictive path switching control results all the time.
3. The voting base predictive path switching control provides a universal predictive path switching controller.
4. The predictors which give better predictions in the MSE sense do not necessarily give better predictive path switching control results.
5. The ZOC gives a better evaluation than the MSE criterion on the performance of predictors implemented in predictive path switching control.

CHAPTER IX

VOICE QUALITY CONTROL THROUGH PREDICTIVE PATH SWITCHING

In this chapter, the behavior of the network is emulated, a real voice signal is encoded into VoIP packets and is sent through the emulated network, the voting based predictive path control is implemented and used to control the VoIP flow. The VoIP packets received following network transport are decoded back into voice, and the quality of the received voice is tested.

A. Test Procedure

A system block diagram for a full-duplex implementation of a VoIP application with predictive path switching control is given in Fig. 63. The two sides of the system are peers, they are using the same VoIP application and the same predictive path switching controllers. Without loss of generality, take User A talking to User B, for example. VoIP application A at User A side collects sufficient voice signal of User A for a VoIP packet, encodes it into frames, packetizes the frames into a VoIP packet, and passes the VoIP packet to the predictive path switching controller A at User A side. In the mean time, controller A transmits probing packets over both the forward path of Path A and the forward path of Path B. The probing packets from controller A are received by controller B at User B side, and are piggybacked to User A with the probing packets from controller B. The probing packets from controller B are transmitted over the backward path of Path A and the backward path of Path B, and are received by controller A. The probing packets from controller B are piggybacked with the next probing packets from controller A. The piggybacked probing packets of controller A are separated and are sent to the information extraction block. The measurements are extracted in the information extraction block and are sent to the

prediction block. In the prediction block, predictions are made based on the measurements and are fed into the control decision block. The control decision block chooses the path to use and transmit the VoIP packets over that path. The VoIP packets from User A are received by User B side controller and are forwarded to the playback buffer of the VoIP application B at the User B side. Then the VoIP packets are decoded and played back at the User B side. For voice quality test, only half of the full system is implemented, and system block diagram is given in Fig. 64.

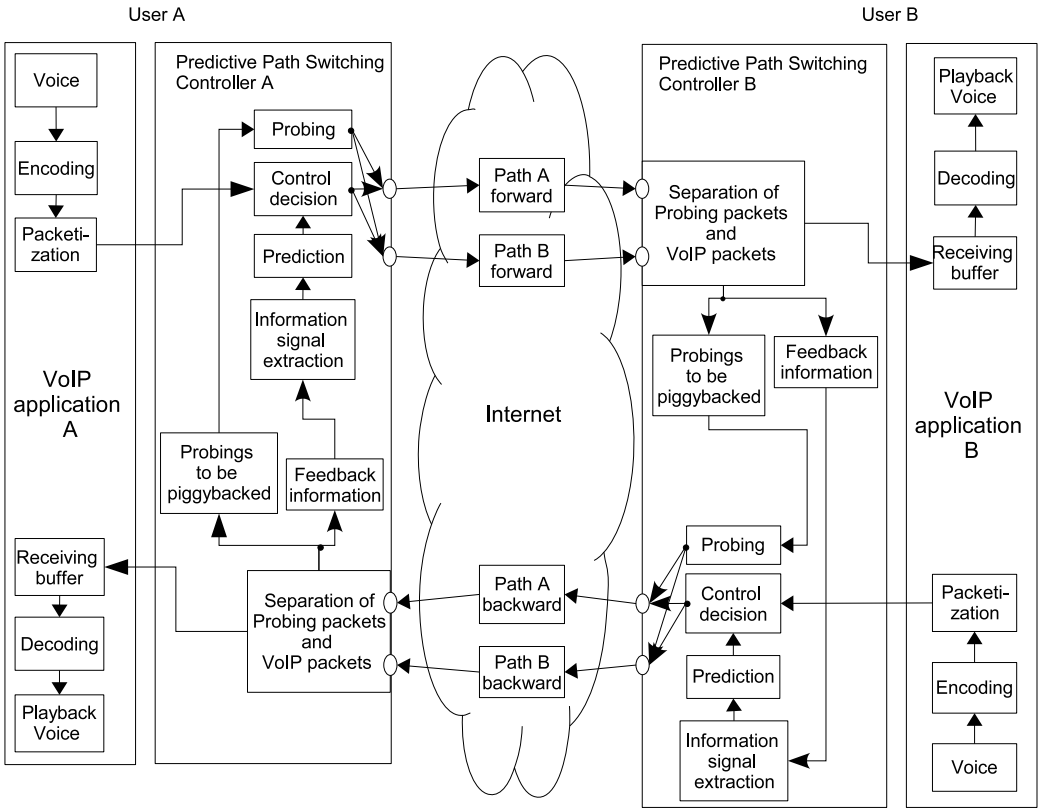

Fig. 63. Full-duplex system block diagram of a VoIP application with predictive path switching control.

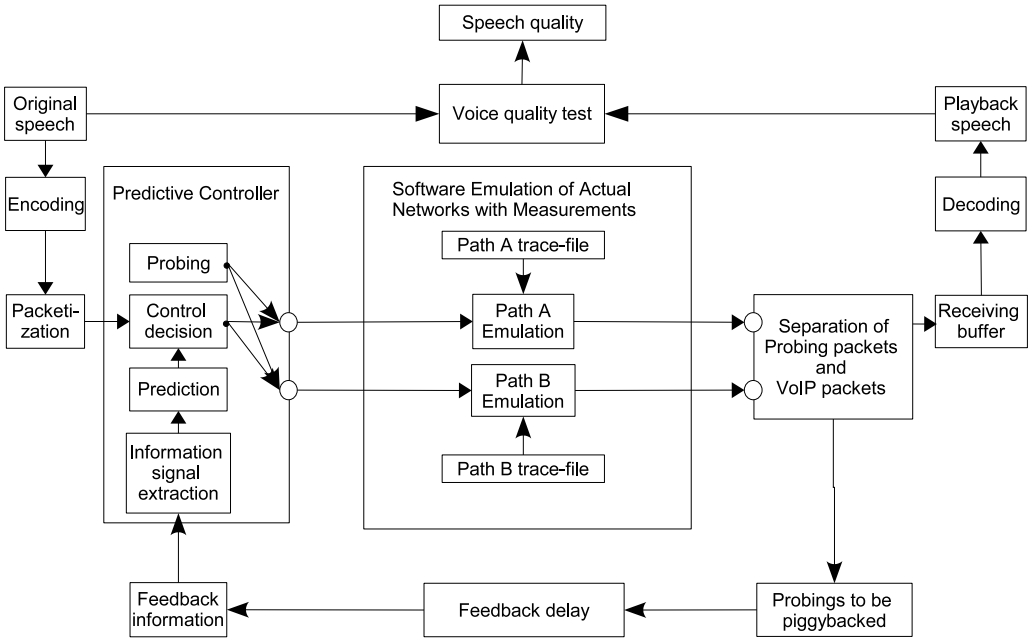

Fig. 64. System block diagram for voice quality test.

1. Packetization of the voice signal

The voice file used in this study is obtained from the CD of *The Fires of Heaven* [165] by Robert Jordan. The *Speex* [93] codec is used to encode the speech. The frame size of the Speex codec is 20 ms. Five frames are packetized into one packet. So the inter departure time of the VoIP packets are 100 ms.

2. Emulation of the network

The behavior of the network is emulated using software and actual network data. Take switching between Path A and Path B, for example. The software emulator reads the trace-files of Path A and Path B, and applies the delays and losses from the trace-file to the VoIP packets according to their send time and over which path they are transmitted. The delays and losses from the trace-files are also applied

to the probing packets according to their send time and over which path they are transmitted.

3. Implementation of the controller

The controller is implemented using three blocks. The probing block, the prediction block, and the control block.

a. The probing block

In the probing block, on the sender side probing packets are sent to both Path A and Path B every 100 ms of emulation time. After the probing packets are assigned delays or are marked as lost by the networks emulator, the probing packets are held at the receiver side until the information feedback limit of 400 ms. Then the probing packets are given to the feedback information extraction function to calculate the information signals such as CLR signals and delay signals. Those packets which have experienced delays more than the information feedback limit of 400 ms are treated as lost packets by the feedback information extraction function. The extracted information signals are averaged every 400 ms (4 probing packets) and fed into the prediction block.

b. The prediction block

All types of predictors, including SP, AR, NAR, RBF, and ad hoc predictors, are implemented in this block. The predictor parameters and coefficients for predicting CLR signals and delay signals are loaded. With the feedback information signals extracted from the probing block, this block provides the two-step-ahead predictions of the information signals given by each type of predictors.

c. The control block

SP_{CLR} based predictive path switching control and the voting based predictive path switching control are implemented in this block. Taking the predictions from the predictor block, this block decides the best path and transmits the VoIP packets over that path.

If the information feedback delay limit is not applied in the probing block, then the controller works using instant feedback, one-step-ahead prediction instead of two-step-ahead prediction, and the results of voting based one-step-ahead predictive path switching control are obtained. This emulates the case when there is no feedback delay.

If the information is directly extracted from the trace-files ahead of time instead of extracting from the probing packets, the ideal case path switching control results are obtained.

4. Voice quality test

In this chapter the PESQ test is used. The C source code for the PESQ test is obtained from ITU-T [140]. The reference voice signal is obtained by decoding the VoIP packets directly without any delays or losses. The results of no switching and transmitting over Path A method (Path A method), no switching and transmitting over Path B method (Path B method), SP_{CLR} based predictive path switching control (SP_{CLR} control), voting based predictive path switching control (V7 control), voting based one-step-ahead predictive path switching control (OSV7 control), and the ideal case path switching control (Ideal control) are obtained. The PESQ-MOS of the resulting voice signals are calculated and compared. As the voice files used by ITU for voice quality tests are usually 8 seconds long, in this study, both the reference

voice files and the degraded voice files are also divided into 8 seconds long segments for the PESQ-MOS calculation. The statistical distribution of the resulting PESQ-MOS using different controls are compared.

B. Control Results

Take the 4th trace-file of path pair AB for example. Table XLII presents the resulting average CLR and PESQ-MOS for the different controls. It is observed that SP_{CLR} control increases (improves) the resulting average PESQ-MOS by 1.27 compared to Path B method, and V7 control increases (improves) the resulting average PESQ-MOS by 0.12 and 1.41 compared to Path A method and Path B method, respectively. But the increments (improvements) are not as much as those of OSV7 control (0.31 and 1.60, respectively) and Ideal control (0.62 and 1.91, respectively).

Table XLII. Average CLR and PESQ-MOS of #4 trace-file path pair AB.

	Path A[1]	Path B[2]	SP_{CLR}[3]	V7[4]	OSV7[5]	Ideal[6]
Avg.[7] CLR (%)	9.22	26.04	6.75	5.62	4.03	2.41
SD[8] CLR (%)	4.09	7.05	2.46	2.31	2.11	1.73
Avg. PESQ-MOS	3.54	2.25	3.52	3.66	3.85	4.16
SD PESQ-MOS	1.11	1.16	0.96	0.92	0.80	0.69

[1] No switching and transmitting over Path A.

[2] No switching and transmitting over Path B.

[3] SP of CLR signals based predictive path switching control.

[4] Voting based predictive path switching control.

[5] Voting based one-step-ahead predictive path switching control.

[6] Ideal case path switching control.

[7] Average.

[8] Standard deviation.

When listening to the resulting voice it is observed that when the PESQ-MOS is above 3.5, the voice quality is pretty good. When the PESQ-MOS is below 3 the voice quality is not good. In between there are noticeable defects in the voice, but still acceptable. Table XLIII shows the voice quality distribution of each control. It shows that SP_{CLR} control is better than Path B method (30.81% bad quality segments vs 72.97% bad quality segments), but is worse compared to Path A method (30.81% bad quality segments vs 26.49% bad quality segments), while V7 control is better than both Path A method (22.16% bad segments vs 26.49% bad segmetns) and Path B method (22.16% bad segments vs 72.97% bad segments). Compared to no switching methods, V7 control reduces the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (14.06% bad segments). The Ideal control gives the best result (8.11% bad segments).

Table XLIII. Voice quality distribution of #4 trace-file path pair AB.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
Path A	26.49	11.08	62.43
Path B	72.97	7.30	19.73
SP_{CLR}	30.81	15.41	53.78
V7	22.16	15.41	62.43
OSV7	14.06	12.97	72.97
Ideal	8.11	4.59	87.30

Table XLIV gives the percentage of segments that have voice quality below 3 for each control, the percentage of segments that have voice quality below 3.5 for each control, the percentage improvement of the predictive path switching controls

over the no switching controls in terms of the percentage of segments with voice quality below a certain level, and the percentage improvement of other predictive path switching controls over the SP based predictive path switching control in terms of the percentage of segments with voice quality blow a certain level. The table shows that the percentage improvements of SP_{CLR} control and V7 control over Path B method are positive, they improve the voice quality compared to Path B method. But the percentage improvement of SP_{CLR} control over Path A method is negative, it is doing worse than Path A method. The percentage of PESQ-MOS below 3 shows that V7 control reduces the percentage of bad segments compared to Path A method. But the percentage of PESQ-MOS below 3.5 shows that V7 control only matches Path A method for the percentage of no more than fair segments. It is very difficult to tell whether Path A or Path B is a better path, when transmitting packets over the network without switching path. So at least the V7 method is a consistent method which is no worse than no switching. The OSV7 control results indicate that if there is no feedback delay, then the predictive path switching control is always better than no switching. The ideal case predictive path switching control gives the best results.

Table XLV is the comparison of the predictive path switching control with the no switching. It gives the percentage of voice segment that has been improved or degraded. When the predictive path switching control result PESQ-MOS is within 0.05 of the results of no switching, it is called the "same", otherwise it is called "better" or "worse" depending on whether the resulting PESQ-MOS is higher or lower than the results of no switching methods. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same; there is hardly any case of worse performance than no switching control. But because of the information feedback delay and prediction error, there are segments where other predictive path switching controls make voice quality worse. More than 70% of the

Table XLIV. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #4 trace-file path pair AB.

	Path A	Path B	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	26.49	72.97	30.81	22.16	14.05	8.11
Improvement over Path A (%)			-16.31	16.35	46.96	69.38
Improvement over Path B (%)			57.78	69.63	80.75	88.89
Improvement over SP_{CLR} (%)				28.08	54.40	73.68
PESQ-MOS < 3.5 (%)	37.57	80.27	46.22	37.57	27.03	12.70
Improvement over Path A (%)			-23.02	0.00	28.05	66.20
Improvement over Path B (%)			42.42	53.20	66.33	84.18
Improvement over SP_{CLR} (%)				18.71	41.52	72.52

times SP_{CLR} control and V7 control results are better than Path B method results. Around 17% of the times SP_{CLR} control and V7 control results are worse than Path B method results. So overall SP_{CLR} control and V7 control are better than Path B method. Also, 22.70% of the times V7 control results are better than Path A method results, and 17.57% of the times V7 control results are worse than Path A method results. So overall V7 control is also better than Path A method.

Table XLV. Comparison of predictive path switching control and no switching methods for #4 trace-file path pair AB.

	Comparing to Path A (%)			Comparing to Path B (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	20.00	48.92	31.08	73.24	9.46	17.30
V7	22.70	59.73	17.57	76.75	7.03	16.22
OSV7	32.97	57.84	9.19	81.62	7.30	11.08
Ideal	50.27	48.92	0.81	87.84	11.89	0.27

The PESQ-MOS distributions calculated over every 0.5 MOS point are presented in Table XLVI. A zoom-in version is plotted in Fig. 65. From the distribution and the plots, it can be seen that the predictive path switching controls reduce the number of extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But the control cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table XLVI. PESQ-MOS distributions calculated over every 0.5 MOS point for #4 trace-file path pair AB.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A	8.65	5.14	4.59	8.11	11.08	14.86	47.57
Path B	36.76	14.05	10.27	11.89	7.30	8.38	11.35
SP_{CLR}	3.51	3.78	10.00	13.51	15.41	12.70	41.09
V7	2.97	2.70	6.22	10.27	15.41	15.68	46.75
OSV7	1.62	1.89	3.51	7.03	12.97	18.11	54.87
Ideal	1.62	0.54	0.81	5.14	4.59	10.81	76.49

The cumulative PESQ-MOS distribution is plotted in Fig. 66. It shows the percentage of voice segment below a given MOS level for each control. Below MOS level of 2, the plots of predictive path switching controls are lower (better) than the no switching methods. At MOS level of 2.5 plot of SP_{CLR} control matches that of Path A method, and above that MOS level the plot of SP_{CLR} control is higher (worse) than Path A method. The plot of V7 control matches with Path A method after

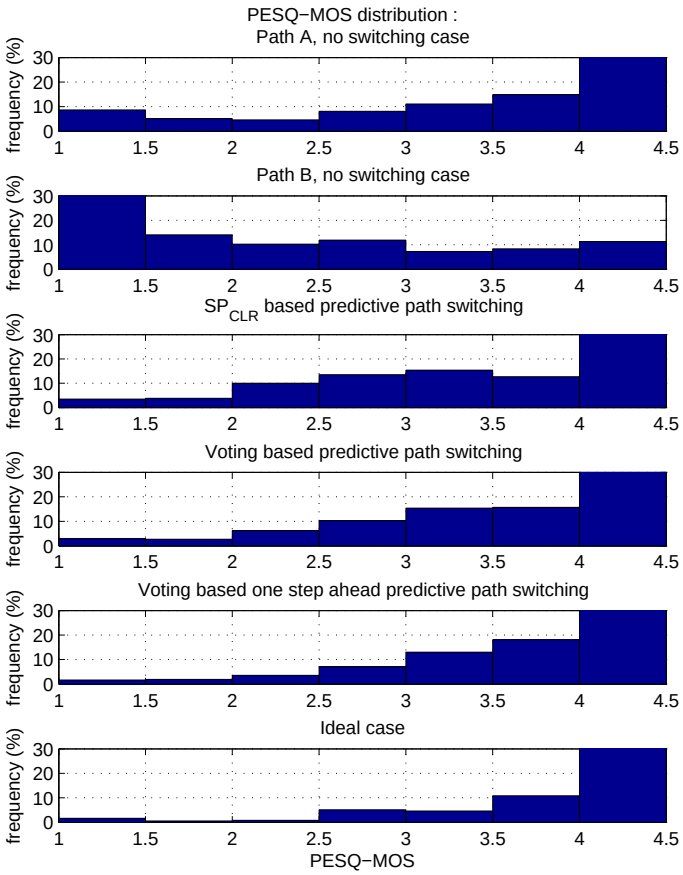

Fig. 65. Zoom-in PESQ-MOS distribution of #4 trace-file path pair AB.

MOS level of 3, which indicates V7 control improves the bad quality segments (below MOS level of 2) to the fair quality range (around MOS level of 3), but has difficulty in improving them above the fair quality level. The plot of OSV7 control is lower (better) than SP_{CLR} control and V7 control. The plot of Ideal control is the lowest (best).

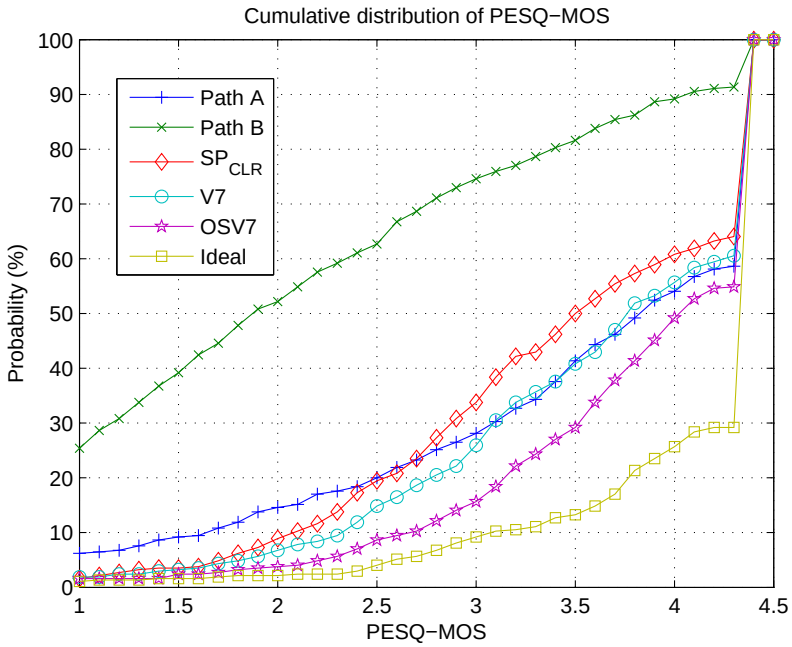

Fig. 66. Cumulative PESQ-MOS distribution of #4 trace-file path pair AB.

The voice quality tests are also performed on the 8^{th} trace-file of path pair AB, and the 4^{th} and 8^{th} trace-files of other path pairs. These results are included in Appendix A, Tables LIII through LXXXVII, and Figs. 73 through 86. Similar results as above are obtained.

C. Chapter Summary and Conclusions

In this chapter, a voice quality test for the predictive path switching controllers is performed. The results show that:

1. Compared to the no switching methods, the predictive path switching controls improve the average resulting PESQ-MOS.

2. Through the distribution and quality classification of the PESQ-MOS results of all the speech segments, it is shown that predictive path switching controls significantly reduce the number of very bad quality segments, and manage to improve the originally bad quality segments to become fair quality segments. But they have difficulty in improving the bad quality segments to become good quality segments.

3. Comparing the results of the two-step-ahead predictive path switching control to the no feedback delay one-step-ahead predictive path switching control and the ideal case path switching control, it is shown that the prediction errors of the information signals and the information feedback delay in the network are the main reasons that hamper the predictive path switching control.

CHAPTER X

IMPLEMENTATION ASPECTS OF PREDICTIVE PATH SWITCHING CONTROL

In previous chapters, a different set of predictors is developed for each different path; this becomes non-scalable when more paths are taken into consideration. The first topic in this chapter is to investigate the possibility of using only one unified set of predictors for all available paths.

The previous chapters also show that both reducing the path switching interval and increasing the predictor complexity can reduce the resulting CLR and thus improve the resulting quality of the real-time multimedia applications. But to reduce the path switching interval, the path probing rate must be increased, which will increase the overhead in traffic caused by the controller. Also the use of complex predictors requires more computational power, which may increase the system implementation cost. The second topic in this chapter is the trade-off between the path switching interval (or the path probing rate), the prediction complexity, and the resulting voice quality.

Another implementation problem that is discussed in this chapter is the investigation of the benefits and drawbacks of using more than two paths in predictive path switching control.

A. Unified Predictors across All Available Paths

The same data sets used in previous chapters are used here as well. To explore the impact of using a unified set of predictors for the prediction of information signals for all path pairs in predictive path switching control, the following test is done. Without loss of generality, take path pair AB for example, and instead of using the

corresponding sets of predictors for the prediction of information signals for Path A and Path B, the set of predictors developed for Path A information signals prediction is used for the the prediction of information signals for both paths. The voting based predictive path switching control results in terms of CLR are placed in the row named $V7_A$ in Table XLVII. The test is conducted for the 9 pairs of one hour long trace-files for path pair AB, and the average of the resulting CLR is placed in the last column.

Similarly the predictor sets developed for Path B and Path C are used for the prediction of both information signals of Path A and Path B as well. The predictive path switching control results in terms of CLR are placed in the rows $V7_B$ and $V7_C$ in Table XLVII, respectively. For comparison, results in terms of CLR for no switching, simple predictors for CLR signals based predictive path switching control (SP_{CLR}), and the voting based predictive path switching control (V7), are given in Table XLVII as well. The same unified predictor tests are also conducted for path pairs AC, $A_{new}B_{new}$, and $A_{new}C_{new}$. The results of the predictive path switching controls are plotted in Fig. 67 for comparison.

From Fig. 67(a), it can be seen that for switching between path pair AB the results in terms of CLR for all the voting based predictive path switching controls with unified predictors ($V7_A$, $V7_B$, and $V7_C$ controls) are worse (higher) than the results in terms of CLR of the normal one (V7 control). The results in terms of CLR are close to the SP_{CLR} control. In Fig. 67(c), $V7_B$ control which uses the predictors developed for Path B is the worst, $V7_A$ control and $V7_C$ control which use the predictors developed for Path A and Path C, respectively, are better, but are still not as good as V7 control which uses different predictors for different paths. These results are enough to prove that it is not possible to use a unified set of predictors in predictive path switching control for all path pairs. At least not using the prediction schemes used in this research.

Table XLVII. Results of voting based predictive path switching control with unified predictors.

Path pair	Method	Resulting CLR on trace-file (%)									Avg. (%)
		1	2	3	4	5	6	7	8	9	
AB	Path A	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89
	Path B	25.21	15.59	18.54	26.23	19.53	15.75	18.83	18.47	25.35	20.39
	SP_{CLR}	9.11	4.48	5.21	6.85	5.83	6.07	6.31	5.56	5.02	6.05
	V7	8.36	3.51	4.50	5.69	5.05	5.43	5.49	4.88	4.23	5.24
	$V7_A$[1]	8.93	4.83	5.21	6.38	5.83	6.17	6.36	5.92	5.31	6.11
	$V7_B$[2]	8.98	4.91	5.29	6.59	6.01	6.32	6.52	6.07	5.69	6.27
	$V7_C$[3]	9.16	4.87	5.21	6.54	5.90	6.35	6.41	6.04	5.51	6.22
AC	Path A	18.54	6.41	8.36	9.15	7.50	10.53	10.39	14.78	12.38	10.89
	Path C	17.45	4.76	7.37	7.14	9.42	5.02	8.66	6.15	14.50	8.94
	SP_{CLR}	4.82	2.55	3.52	3.27	3.05	1.82	3.27	3.77	3.25	3.26
	V7	4.71	2.06	3.01	2.99	2.45	1.73	2.87	3.54	3.58	3.00
	$V7_A$	4.65	2.02	3.01	3.09	2.64	1.70	2.88	3.46	3.35	2.98
	$V7_B$	4.84	1.96	3.08	3.12	2.55	1.99	3.11	3.48	3.43	3.06
	$V7_C$	4.71	2.10	3.03	3.07	2.41	1.86	2.88	3.49	3.27	2.98
$A_{new}B_{new}$	Path A_{new}	8.70	8.95	7.11	5.48	7.91	6.91	5.07	6.32	11.22	7.52
	Path B_{new}	12.18	12.84	23.39	13.31	8.88	28.70	33.48	24.08	8.99	18.43
	SP_{CLR}	2.89	2.72	4.53	2.02	3.35	3.79	3.31	2.55	4.15	3.26
	V7	2.79	2.58	4.36	1.90	2.97	3.39	2.89	2.48	4.46	3.09
	$V7_A$	2.93	2.78	4.47	2.10	2.99	3.49	2.93	2.58	4.13	3.16
	$V7_B$	3.30	3.40	4.62	2.40	3.23	4.10	3.15	3.10	4.33	3.51
	$V7_C$	3.06	2.93	4.47	2.26	3.08	3.61	2.96	2.81	4.17	3.26
$A_{new}C_{new}$	Path A_{new}	8.70	8.95	7.11	5.48	7.91	6.91	5.07	6.32	11.22	7.52
	Path C_{new}	4.24	4.24	4.16	4.58	10.88	4.13	4.74	7.04	4.89	5.43
	SP_{CLR}	1.38	2.01	1.50	1.60	2.82	1.39	1.67	2.01	2.14	1.84
	V7	1.28	1.57	1.11	1.49	2.62	1.23	1.54	1.78	1.74	1.60
	$V7_A$	1.24	1.68	1.18	1.37	2.44	1.34	1.59	1.78	1.84	1.61
	$V7_B$	1.27	1.61	1.32	1.42	2.66	1.45	1.53	1.90	2.10	1.69
	$V7_C$	1.25	1.72	1.20	1.46	2.64	1.40	1.48	1.78	1.94	1.65

[1] Voting based predictive path switching control using predictors of Path A for prediction of all other paths.

[2] Voting based predictive path switching control using predictors of Path B for prediction of all other paths.

[3] Voting based predictive path switching control using predictors of Path C for prediction of all other paths.

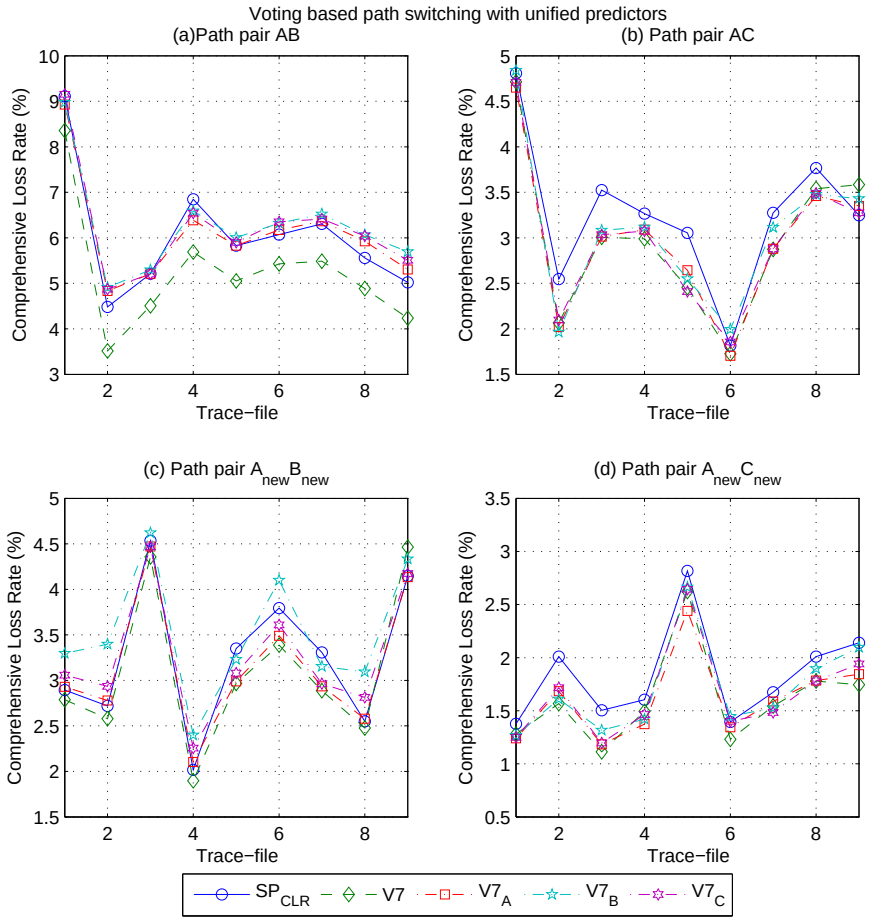

Fig. 67. Results of voting based predictive path switching with unified predictors in terms of CLR. (a) Results on path pair AB. (b) Results on path pair AC. (c) Results on path pair $A_{new}B_{new}$. (d) Results on path pair $A_{new}C_{new}$.

Yet in Fig. 67(b) and (d), it is interesting to see that the predictive path switching controls with unified predictors are acceptably good. Their results in terms of CLR are much closer to those of the predictive path switching control with non-unified predictors in these two cases. The results in terms of CLR for predictive path switching control with predictor sets developed for Path A and Path C are better than the results in terms of CLR of the predictive path switching control with the set of predictors developed for Path B. Recall that the average CLR for Path A is 11%, which is close to that of Path C (9%), and is very different from that of Path B (20%), and the same holds for Path A_{new} (7.5%), B_{new} (18.4%), and C_{new} (5.4%). So a reasonable guess would be that a unified set of predictors can be used for the prediction of the information signals for paths within the same CLR range. Separate sets of predictors are needed for paths with different CLR ranges. As for paths A, B, C, A_{new}, B_{new}, and C_{new}, one set of predictors is needed for predicting the paths with CLR around 10%, and another set of predictors are need for the paths with CLR around 20%.

The test results in terms of E-model MOS are given in Table XLVIII for path pairs AB, AC, $A_{new}B_{new}$, and $A_{new}C_{new}$. The E-model MOS results are also plotted in Fig. 68 for comparison. The same conclusion can be draw from the E-model MOS results. It is not possible to use only one set of predictors for the prediction of the information signals of all the paths, but it is possible to use one set of predictors for the prediction of the information signals of the paths with CLRs in the same range. Therefore, some reduction in the number of developed predictors can be achieved based on the average CLR of a path.

Table XLVIII. Results of voting based predictive path switching control with unified predictors in terms of E-model MOS.

Path pair	Method	Resulting MOS on trace-file (%)									Avg.
		1	2	3	4	5	6	7	8	9	
AB	Path A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82
	Path B	1.00	1.43	1.19	1.00	1.19	1.46	1.22	1.26	1.00	1.19
	SP_{CLR}	2.14	2.86	2.74	2.45	2.63	2.60	2.56	2.69	2.77	2.60
	V7	2.23	3.03	2.86	2.63	2.75	2.70	2.69	2.80	2.90	2.73
	$V7_A$	2.17	2.81	2.75	2.53	2.64	2.59	2.56	2.63	2.73	2.60
	$V7_B$	2.16	2.79	2.73	2.49	2.61	2.57	2.53	2.61	2.66	2.57
	$V7_C$	2.14	2.80	2.75	2.50	2.62	2.57	2.55	2.61	2.69	2.58
AC	Path A	1.10	2.46	2.17	2.02	2.29	1.74	1.88	1.24	1.50	1.82
	Path C	1.10	2.82	2.36	2.41	2.03	2.77	2.10	2.56	1.52	2.19
	SP_{CLR}	2.81	3.21	3.03	3.08	3.12	3.36	3.08	2.98	3.08	3.08
	V7	2.83	3.31	3.13	3.13	3.23	3.38	3.15	3.03	3.02	3.13
	$V7_A$	2.84	3.32	3.13	3.11	3.20	3.38	3.15	3.04	3.06	3.14
	$V7_B$	2.80	3.33	3.11	3.11	3.21	3.32	3.11	3.04	3.05	3.12
	$V7_C$	2.83	3.30	3.12	3.12	3.24	3.35	3.15	3.03	3.08	3.14
$A_{new}B_{new}$	Path A_{new}	1.99	1.99	2.35	2.64	2.28	2.40	2.75	2.50	1.80	2.30
	Path B_{new}	1.61	1.65	0.99	1.49	2.05	1.00	1.00	1.00	2.06	1.43
	SP_{CLR}	3.14	3.17	2.82	3.32	3.06	2.98	3.06	3.21	2.92	3.08
	V7	3.16	3.20	2.86	3.34	3.14	3.05	3.15	3.23	2.87	3.11
	$V7_A$	3.13	3.16	2.84	3.30	3.13	3.04	3.14	3.21	2.92	3.10
	$V7_B$	3.06	3.05	2.81	3.24	3.09	2.93	3.09	3.11	2.89	3.03
	$V7_C$	3.11	3.14	2.84	3.27	3.11	3.01	3.13	3.16	2.92	3.08
$A_{new}C_{new}$	Path A_{new}	1.99	1.99	2.35	2.64	2.28	2.40	2.75	2.50	1.80	2.30
	Path C_{new}	2.90	2.90	2.92	2.85	1.89	2.92	2.80	2.41	2.78	2.71
	SP_{CLR}	3.45	3.32	3.42	3.40	3.16	3.44	3.39	3.32	3.29	3.35
	V7	3.47	3.41	3.50	3.42	3.20	3.48	3.41	3.37	3.37	3.40
	$V7_A$	3.48	3.38	3.49	3.45	3.24	3.46	3.40	3.37	3.35	3.40
	$V7_B$	3.47	3.40	3.46	3.44	3.19	3.43	3.42	3.34	3.30	3.38
	$V7_C$	3.47	3.38	3.48	3.43	3.20	3.44	3.43	3.37	3.33	3.39

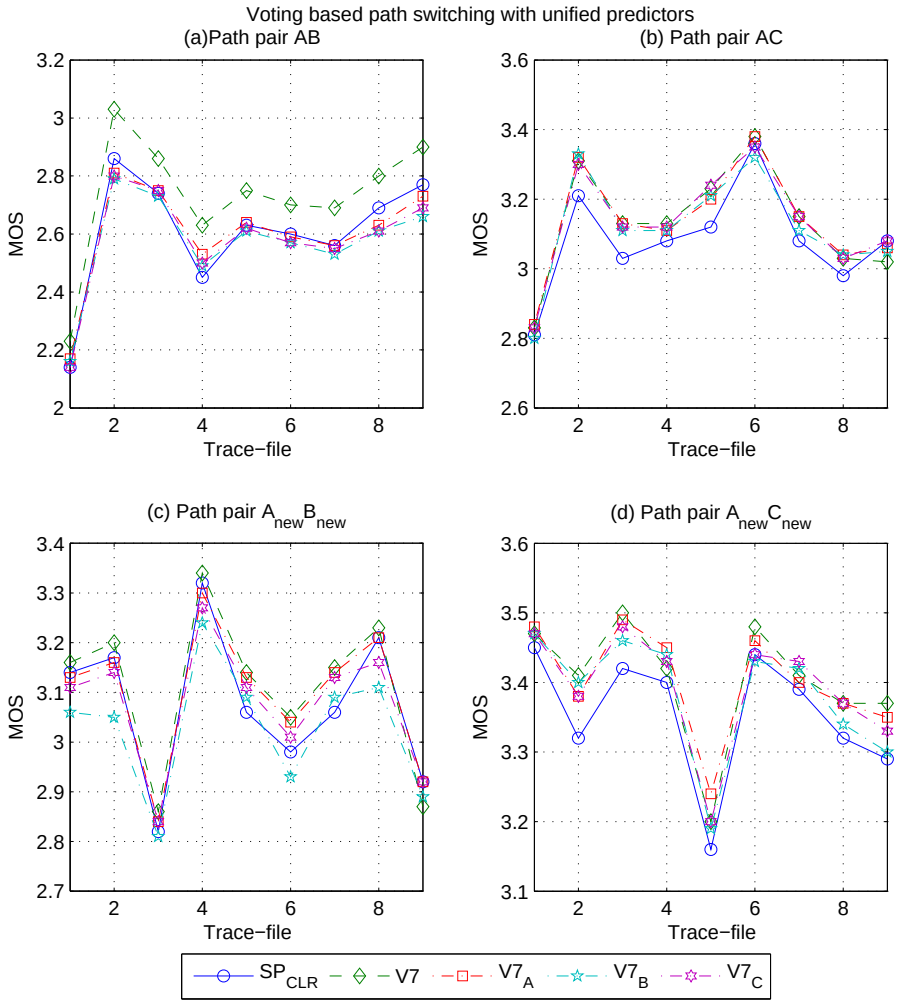

Fig. 68. Voting based path switching with unified predictors in terms of E-model MOS. (a) Results on path pair AB. (b) Results on path pair AC. (c) Results on path pair $A_{new}B_{new}$. (d) Results on path pair $A_{new}C_{new}$.

B. Trade-off between Probing Rate, Prediction Complexity, and Resulting Voice Quality

In previous chapters, it is shown that both reducing the path switching interval and increasing the prediction complexity can improve the resulting voice quality of interactive multimedia applications. But the use of complex predictors will require more computational power, and reduction of the path switching interval, will increase the probing rate, which will increase the overhead in probing traffic.

For example, when probing two path every 100 ms, a real implementation of the probing packets has the following structure:

```
typedef struct _msg_small
{
    USHORT seqNo;
    USHORT bytesSent;
    UINT t_sent;
    USHORT seqNoEcho;
    USHORT bytesRcvd;
    UINT t_send_echo;
    UINT t_rcvd_echo;
} SMALLMSGHEADER;
```

where *seqNo* is the sequence number of the packet; *bytesSent* is the total number of bytes sent, which is used for accumulation calculation; *t_sent* is the send time stamp of the packet; *seqNoEcho* is the sequence number of the piggybacked probing packet from the other side; *bytesRcvd* is the total number of bytes received which is used in accumulation calculation for the other side; *t_send_echo* is the send time stamp of the

piggybacked probing packet from the other side; and *t_rcvd_echo* is the receive time stamp of the piggybacked probing packet from the other side. This header is 20 bytes long. Plus the 8 bytes of the UDP header, and 20 bytes of the IP header, results in a packet with a header of 48 bytes. So if probing every 100 ms, i.e. a probing rate of 10 Hz, an overhead in probing traffic of almost 3.84 kbps results in each path. If the path switching interval is reduced to 20 ms, i.e. a probing rate of 50 Hz, this will generate an overhead in probing traffic of 19.2 kbps in each path. For path switching between two paths, the total overhead in probing traffic will be 38.4 kbps, which is almost as high as running another VoIP application. On the other hand if the probing rate is dropped to 2.5 Hz, i.e. switching every 400 ms, the overhead traffic is 0.96 kbps in each path, which is reduced significantly.

1. Test with resampled probing

To explore the impact of the trade-off between probing rate, predictor complexity, and resulting CLR, the following predictive path switching controls are tested. In the first one, named SP_{100}, the paths are probed every 100 ms, and the path switching is also performed every 100 ms. A five-step-ahead SP of CLR signals is used in this case. In the second one, named SP_{400}, the paths are probed every 400 ms, and the path switching is performed every 400 ms. A two-step-ahead SP for CLR signals is used. In the third one, named $V7_{400}$, the paths are probed every 400 ms, the path switching is performed every 400 ms, and the voting based predictive path switching control is used. In all these controls, the VoIP packets are all sent every 100 ms. As the trace-files are obtained with 100 ms probing, for the 400 ms probing cases, resampling is used. The test results in terms of CLR are given in Table XLIX and plotted in Fig. 69.

From the results it can be seen that the faster probing rate gives better predic-

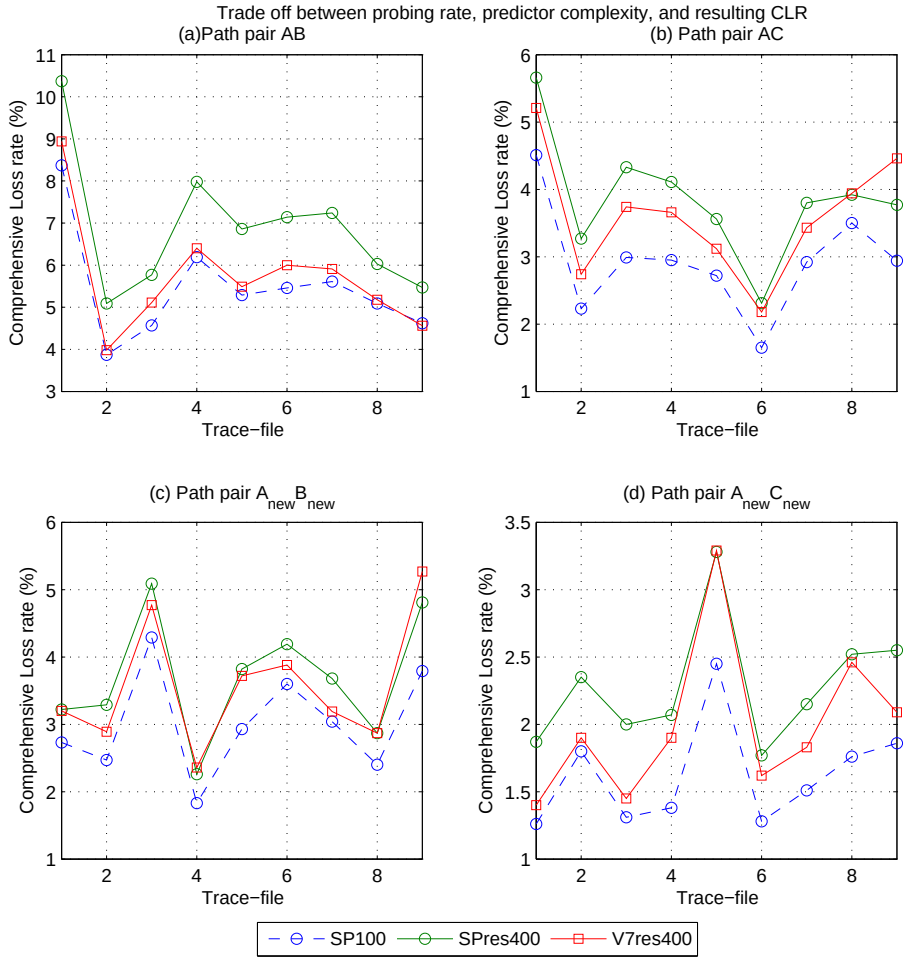

Fig. 69. Trade-off between probing rate, predictor complexity, and resulting CLR. (a) Results on path pair AB. (b) Results on path pair AC. (c) Results on path pair $A_{new}B_{new}$. (d) Results on path pair $A_{new}C_{new}$.

Table XLIX. Trade-off between probing rate, predictor complexity, and resulting CLR.

Path pair	Method	Prob. rate	Resulting CLR on trace-file (%)									Avg.
		(kbps/path)	1	2	3	4	5	6	7	8	9	(%)
AB	SP_{100}	3.84	8.37	3.87	4.57	6.19	5.29	5.46	5.61	5.09	4.62	5.45
	SP_{400}	0.96	10.37	5.09	5.77	7.98	6.86	7.14	7.24	6.03	5.47	6.88
	$V7_{400}$	0.96	8.94	3.98	5.11	6.40	5.49	6.00	5.91	5.18	4.56	5.73
AC	SP_{100}	3.84	4.51	2.23	2.99	2.95	2.72	1.65	2.92	3.50	2.94	2.93
	SP_{400}	0.96	5.66	3.27	4.33	4.11	3.56	2.31	3.80	3.92	3.77	3.86
	$V7_{400}$	0.96	5.21	2.74	3.74	3.66	3.12	2.18	3.43	3.94	4.46	3.61
$A_{new}B_{new}$	SP_{100}	3.84	2.73	2.47	4.29	1.83	2.93	3.60	3.04	2.40	3.79	3.01
	SP_{400}	0.96	3.22	3.29	5.09	2.26	3.82	4.19	3.68	2.87	4.81	3.69
	$V7_{400}$	0.96	3.20	2.89	4.77	2.36	3.72	3.88	3.19	2.87	5.27	3.57
$A_{new}C_{new}$	SP_{100}	3.84	1.26	1.80	1.31	1.38	2.45	1.28	1.51	1.76	1.86	1.62
	SP_{400}	0.96	1.87	2.35	2.00	2.07	3.28	1.77	2.15	2.52	2.55	2.29
	$V7_{400}$	0.96	1.40	1.90	1.45	1.90	3.29	1.62	1.83	2.46	2.09	2.00

tive path switching control results in terms of CLR. The SP based predictive path switching control with probing interval of 100 ms gives an average improvement of 0.5 percentage point in terms of the CLR over the voting based predictive path switching control with probing interval of 400 ms. On the other hand it has three times as much overhead in probing traffic. The $V7_{400}$ control is more complicated than the SP_{400} control which has the same probing interval of 400 ms, but on average it is 0.5 percentage point better in terms of the CLR.

The test results in terms of E-model MOS are given in Table L, and plotted in Fig. 70. From the results in terms of E-model MOS, the SP_{100} control gives an average improvement of around 0.08 in terms of E-model MOS compared to the $V7_{400}$ control. $V7_{400}$ control gives an average improvement of around 0.05 in terms of E-model MOS compared to SP_{400} control which has the same probing interval of 400 ms. On the other hand the fast probing control has three times as much overhead in probing

Table L. Trade-off between probing rate, predictor complexity, and resulting MOS.

Path pair	Method	Prob. rate	Resulting MOS on trace-file									Avg.
		(kbps/path)	1	2	3	4	5	6	7	8	9	
AB	SP_{100}	3.84	2.24	2.97	2.85	2.55	2.72	2.70	2.68	2.76	2.84	2.70
	SP_{400}	0.96	1.97	2.75	2.64	2.27	2.46	2.42	2.41	2.60	2.69	2.47
	$V7_{400}$	0.96	2.15	2.95	2.75	2.51	2.67	2.60	2.61	2.74	2.84	2.65
AC	SP_{100}	3.84	2.86	3.27	3.13	3.14	3.18	3.39	3.14	3.03	3.14	3.14
	SP_{400}	0.96	2.66	3.08	2.89	2.93	3.02	3.26	2.98	2.96	2.98	2.97
	$V7_{400}$	0.96	2.74	3.18	2.99	3.01	3.11	3.29	3.05	2.95	2.87	3.02
$A_{new}B_{new}$	SP_{100}	3.84	3.17	3.22	2.87	3.35	3.14	3.01	3.11	3.24	2.98	3.12
	SP_{400}	0.96	3.07	3.07	2.72	3.27	2.98	2.90	3.00	3.15	2.81	3.00
	$V7_{400}$	0.96	3.08	3.14	2.78	3.25	3.00	2.97	3.09	3.15	2.74	3.02
$A_{new}C_{new}$	SP_{100}	3.84	3.47	3.36	3.46	3.45	3.23	3.47	3.42	3.37	3.35	3.40
	SP_{400}	0.96	3.34	3.25	3.32	3.31	3.08	3.37	3.29	3.22	3.21	3.27
	$V7_{400}$	0.96	3.44	3.34	3.43	3.34	3.07	3.40	3.35	3.23	3.30	3.32

traffic, and the voting based control is more complicated than the SP based controls.

2. Test on a 20 ms probing experiment

Two new sets of trace-files (Path D and Path E) were obtained with 20 ms probing interval in the Spring 2006 from PlanetLab, on the same pair of nodes, *planetlab1.nbgisp.com* and *planetlab1.gti-dsl.nodes.planet-lab.org*. Path D has an average CLR of 12.7%. Path E has an average CLR of 11.7%. The following methods are tested on this pair of paths:

- Path D method: No switching and transmitting over Path D only.
- Path E method: No switching and transmitting over Path E only.
- SP_{20} control: Predictive path switching control with SPs. The CLR signals are calculated every 20 ms. The path switching is performed every 20 ms. A

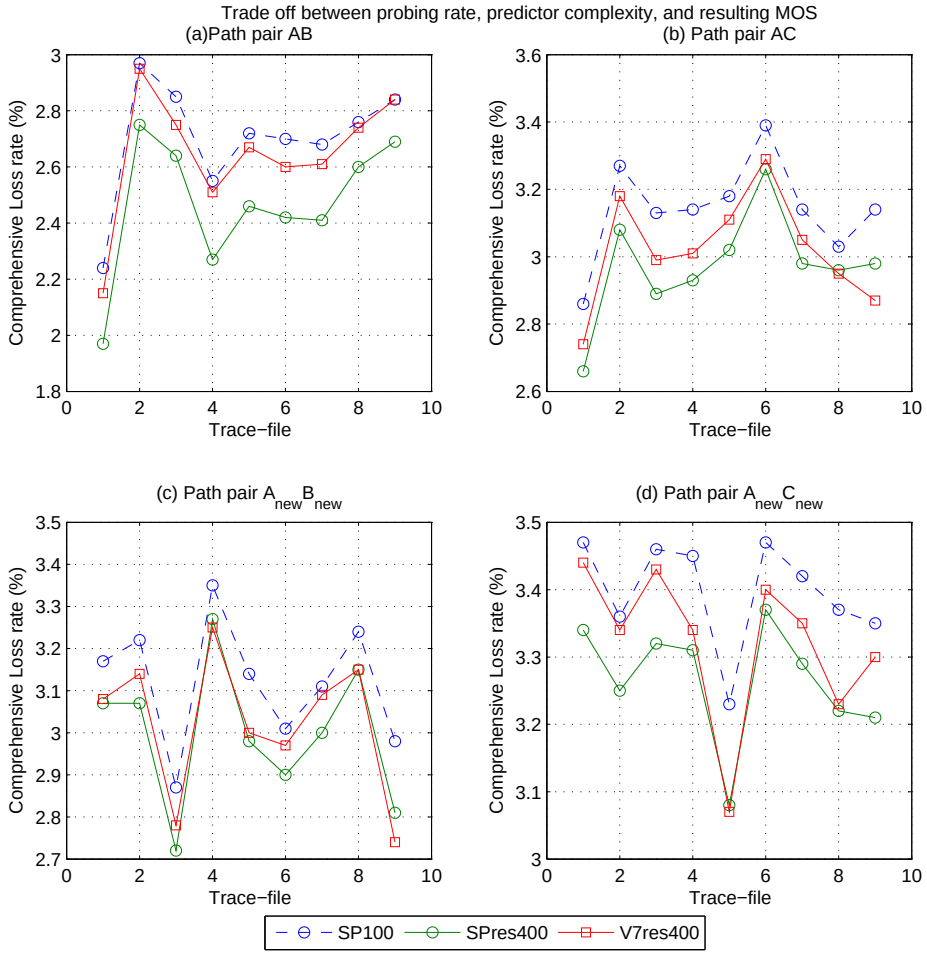

Fig. 70. Trade-off between probing rate, predictor complexity, and resulting MOS. (a) Results on path pair AB. (b) Results on path pair AC. (c) Results on path pair $A_{new}B_{new}$. (d) Results on path pair $A_{new}C_{new}$.

twenty-one-step-ahead SP is used.

- SP_{100} control: Predictive path switching control with SPs. The probing results are resampled at an interval of 100 ms. The CLR signal is calculated based on the resampled probings. The path switching is performed every 100 ms. A five-step-ahead SP is used.

- SP_{400}: Same as SP_{100}, only the resampled probing interval and switching interval is 400 ms. A two-step-ahead SP is used.

- $V7_{400}$: Voting based predictive path switching control. The probing results are resampled at an interval of 400 ms. Both CLR and delay signals are calculated based on the resampled probings. Seven two-step-ahead predictors are used. Path switching is performed every 400 ms.

In these tests the VoIP packets are sent every 20 ms, and for the Speex codec this is one encoded frame per packet. The test results in terms of the CLR are given in Table LI, and plotted in Fig. 71.

Table LI. Trade-off between probing rate, predictor complexity, and resulting CLR on path pair DE.

Path pair	Method	Prob. rate (kbps/path)	Resulting CLR on trace-file (%)									Avg. (%)
			1	2	3	4	5	6	7	8	9	
DE	Path D	0.00	14.57	12.70	5.61	9.44	10.03	11.99	13.47	5.77	30.47	12.67
	Path E	0.00	9.51	7.59	7.25	13.44	20.57	18.75	12.98	7.84	7.27	11.69
	SP_{20}	19.20	2.98	2.30	1.38	2.56	4.66	3.85	3.86	1.44	4.22	3.03
	SP_{100}	3.84	3.19	2.47	1.47	2.66	4.53	4.10	4.06	1.64	4.14	3.14
	SP_{400}	0.96	3.80	3.18	1.60	3.34	5.25	5.15	4.77	2.19	4.51	3.75
	$V7_{400}$	0.96	3.92	2.71	1.87	3.39	5.23	4.83	5.21	2.11	4.43	3.74

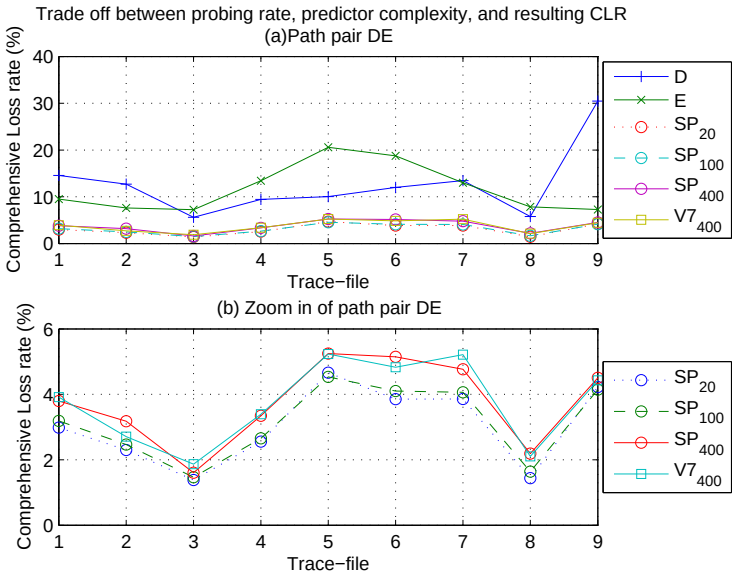

Fig. 71. Trade-off between probing rate, predictor complexity, and resulting CLR for path pair DE.

In these results, still the faster probing rate and path switching rate result in better control results. On average, the SP_{20} control has a 0.7 percentage point improvement in terms of the CLR over SP_{400} and $V7_{400}$ methods, the SP_{100} control has a 0.6 percentage point improvement in terms of the CLR over SP_{400} and $V7_{400}$ controls. Comparing SP_{20} with SP_{100}, on the average there is only 0.1 percentage point improvement in the resulting CLRs, and the four times increment in the probing rate cannot be justified. In this test, the $V7_{400}$ didn't give enough improvement compared to the SP_{400} to justify its prediction complexity.

The test results in terms of the E-model MOS are given in Table LII, and plotted in Fig. 72. In these results, the SP_{20} and SP_{100} controls on the average have about 0.14 improvement in MOS over the SP_{400} and $V7_{400}$ methods. Comparing SP_{20} with SP_{100}, there is only about 0.03 improvement in MOS, and the four times increment

in the probing rate cannot be justified. The $V7_{400}$ does not give enough improvement compared to the SP_{400} control to justify its prediction complexity.

Table LII. Trade-off between probing rate, predictor complexity, and resulting MOS on path pair DE.

Path pair	Method	Prob. rate (kbps/path)	Resulting MOS on trace-file									Avg.
			1	2	3	4	5	6	7	8	9	
DE	Path D	0.00	1.57	1.70	2.48	1.98	1.93	1.74	1.64	2.45	1.00	1.83
	Path E	0.00	1.97	2.19	2.24	1.64	1.27	1.35	1.68	2.16	2.23	1.86
	SP_{20}	19.20	2.99	3.16	3.41	3.09	2.65	2.80	2.80	3.40	2.72	3.00
	SP_{100}	3.84	2.94	3.11	3.39	3.07	2.67	2.75	2.76	3.34	2.74	2.97
	SP_{400}	0.96	2.81	2.95	3.35	2.91	2.54	2.56	2.63	3.19	2.67	2.84
	$V7_{400}$	0.96	2.79	3.05	3.27	2.90	2.55	2.62	2.55	3.21	2.68	2.85

3. Section summary and conclusions

The test results on the six trace-files of paths A, B, C, A_{new}, B_{new}, and C_{new} and the two trace-files of Path D and Path E show that fast probing and switching result in improvements over slow ones. But the improvement in the resulting voice quality might not be enough to justify the increased overhead in probing traffic. The more complex voting based predictive path switching control in most cases is doing better than the simple predictor based controller, but whether or not the increased complexity can be justified by the obtained improvement depends on specific cases. There is always a trade-off between the overhead in probing traffic, the computational complexity, and the resulting voice quality.

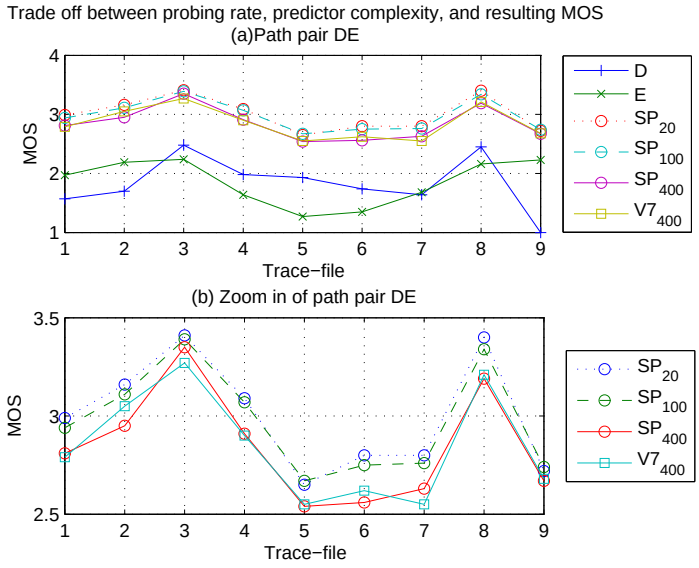

Fig. 72. Trade-off between probing rate, predictor complexity, and resulting MOS for path pair DE.

C. Generalization to Multiple Paths

In previous chapters, it is assumed that there are two paths available in networks between the source and destination pair. This section discusses the possibility that more than two paths might be available.

If there are multiple paths available, then the predictive path switching control logic is almost the same. Instead of picking the best of two paths with the best predicted quality, one must pick the beest among multiple paths.

On one hand, if there are more paths available, then it is more likely to find temporal differences in delays and losses in these paths. At any given time the probability of all the paths having high loss or over-delay is reduced. It is more likely to find a better alternative path to improve the voice quality. So it is expected that the CLR range, where the predictive path switching will be able to improve the voice

quality in a meaningful way, will be enlarged. Especially the high CLR side of this range will be increased. The control result for the multiple paths case is expected to be better than the two paths case for the same average CLR level.

On the other hand, there are significant drawbacks. First, new predictors must be trained for each path in order to use predictive path switching control. The number of predictors increases linearly with the available number of paths. Second, the amount of overhead in probing traffic is increasing linearly with the available number of paths. The volume of the probing traffic will be more than that of the VoIP flow, and controls with high probing rate will become unacceptable.

For the first drawback, if one set of predictors can be used for all the paths with the same range of CLR, then only a limited number of predictors need to be trained. The number of predictors will not increase as fast as the number of available paths.

For the second drawback, if there is more than one VoIP flow going between a given source and destination pair, then one set of probing packets can collect network information for the control of multiple VoIP flows. The more number of VoIP flows being controlled over a given path the more effective probing traffic will become in utilizing network bandwidth. But, if the total amount of VoIP traffic being controlled by the proposed approach is too large, the general assumption that "*the VoIP traffic under control will not change the paths' condition*" may no longer be valid.

D. Chapter Summary and Conclusions

In this chapter, some aspects regarding the implementation of predictive path switching control are investigated. First, the possibility of using a set of unified predictors for all paths in the predictive path switching control is investigated. The results show that it is not possible to use a unified set of predictors in predictive path switching

control for all path pairs. Yet it might still be possible to have a unified set of predictors for the prediction of the information signals of paths within the same CLR range.

Second, the trade-off between probing rate, prediction complexity, and the resulting voice quality is discussed. The results show that the fast probing and switching gives better control results at the cost of more overhead in probing traffic. In most cases, the voting based predictive path switching control is better than the SP based predictive path switching control at increased computational complexity. Whether this complexity can be justified by the improvement obtained will depend on specific cases. The choice of control will be a trade-off of the overhead in probing traffic, computational complexity, and the resulting voice quality.

Finally, the case of more than two available paths is discussed. On one hand this is expected to improve the resulting VoIP QoS. On the other hand there will be some drawbacks such as more implemented predictors and more overhead in probing traffic.

CHAPTER XI

SUMMARY AND CONCLUSIONS

The increasing demand for real-time multimedia applications and the lack of QoS support in public best-effort networks has prompted many researchers to attempt improvements on the QoS of such networks. The router based methods have better control of the networks but are difficult to deploy. The end/edge based methods are easy to deploy but are limited in improvement ability. The overlay based methods come in between. With the increasing availability of path-diversity in public best-effort networks, path-switching control has also received much research interest.

The main aim of this research is to improve the QoS for real-time multimedia applications. The solution should be deployable in a scalable end-to-end manner without any need to change the core infrastructure, and it should be implemented as middleware without changing the existing codecs of the media applications.

This chapter gives the summary and conclusions of this research, presents the contributions and limitation of this work, and finally gives some suggestions for future work.

A. Research Summary

In this research, the assumption behind all approaches is that the flow of interest is relatively small component compared to the other flows in the network and that the delay profile is mostly depended on the cross flow in the network. A continuous fluid model of a signal flow transported over a network system with and without flow reversal is introduced. The fluid model is discretized based on the different assumptions of the input flow. Source buffering based predictive control using this model is investigated.

Predictive path switching control is then introduced. Several concepts used in this approach is discussed. Some emulation studies are performed, exploring the impact of path average comprehensive loss rate and the impact of traffic delay signal frequency content on predictive path switching control.

Actual networks data are collected for further study. Problems that appear in data collection from actual networks are discussed. Some solutions to overcome these problems are given. A preliminary study shows that there is room for QoS improvement through path switching in actual networks.

The minimum required prediction horizon is discussed, and a further study shows that only a limited number of steps ahead of prediction are needed for predictive path switching control. An investigation of the impact of prediction/switching step size on predictive path switching control is conducted.

Five types of predictors: SP, AR, NAR, RBF, and ad hoc, are developed for three different information signals: loss rate, delay, and accumulation. The predictors are tested for their predictive path switching control performances. Then a voting based predictive path switching control is proposed. This control uses the control decision from seven different predictive controllers. A new criterion, ZOC, which provides a better predictive path switching control performance evaluation of the predictors than the MSE is discussed. A voice quality test is performed on a real VoIP segment using the voting based predictive path switching control.

Finally, some implementation aspects of predictive path switching control are investigated. The investigation on the possibility of using a single set of unified predictors for all paths is conducted. The trade-off between probing rate, prediction complexity, and resulting voice quality is investigated. The possibility of using multiple independent paths is discussed.

B. Research Conclusions

Under certain conditions, source buffering based predictive control is proved to be effective for improving the QoS of real-time multimedia applications only when there is sufficient flow reversal in the network. A literature review shows that for the applications of interest there is not enough packet reordering in today's networks for this method to be effective.

The emulation studies results show that the predictive path switching control performs better when the frequency content of the delay signals are low. If the delay signal frequency content is about 0.5 Hz, then the dynamic models based predictors give significantly better predictive path switching control results than the simple predictors. The emulation studies results also show that the predictive path switching control gives the best VoIP QoS improvement when the average CLRs of the two paths are in the 5% to 15% range.

The investigation results of the impact of prediction/switching step size on predictive path switching control show that smaller switching intervals result in more effective control.

The predictive path switching control results using different predictors show that predictive path switching control is generally better than no switching. Yet no one predictor gives the best performances at all times. The ranking of the predictors according to their two-step-ahead predictions in terms of the MSE criterion does not match their ranking according to their control effectiveness. The control results of the voting based predictive path switching control show that the voting scheme gives universally good performance in all tested cases.

The voice quality test results prove that predictive path switching control is better than no switching. Predictive path switching control significantly reduce the

number of very bad quality segments, and manages to improve the bad quality segments to fair quality. But path switching has difficulty in improving the bad quality segments into the good quality zone. The test results also show that the prediction errors in the information signals and the information feedback delay in networks are the main reasons hampering the predictive path switching control performance.

The investigation of using a single set of unified predictors for all paths shows that it is not possible to do that. Yet it is possible to have a unified set of predictors for the prediction of information signals for paths within the same range of CLRs. The discussion on the trade-off between probing rate, prediction complexity, and resulting voice quality shows that faster probing and switching gives better control results at the cost of more overhead in probing traffic. The voting based predictive path switching control is better than a simple predictor based control in most cases, at the cost of more computational complexity. There is a trade-off in the overhead in probing traffic, computational complexity, and resulting voice quality. The discussion on the possibility of using multiple independent paths shows that on one hand it might improves the VoIP QoS, but on the other hand there are some drawbacks such as more predictor training and more overhead in probing traffic.

C. Contributions

This dissertation presents research that has made the following contributions:

1. Single path control: Under certain assumptions, a continuous fluid model of a single flow transported over a best-effort network is developed which allows for flow reversals. This dissertation proves that source buffering based predictive control is effective for improving the QoS of real-time multimedia applications only when there is sufficient flow reversal in the network.

2. Multipath switching control: A voting based predictive path switching control scheme is developed to improve the QoS of real-time multimedia applications. This dissertation demonstrates that predictive path switching can improve the QoS in a meaningful way in actual networks.

D. Limitations

Following are a number of limitations of this research which must be addressed prior to deployment.

1. Limited scalability: If implemented in end nodes, the current control will require one probing flow for each end-to-end path. This is highly non-scalable because of the increasing amount of overhead in probing traffic if multiple paths are taken into consideration and multiple end nodes implement this approach. It may be possible to implement the proposed approach at the edge routers and provide path switching for multiple flows, to solve the scaling problem associated with the overhead in probing traffic. But then, the assumption that the VoIP flow under control is relatively small amount of all flows in the network may be violated, which could significantly change the results.

2. Limited data set: The data used in this research have been collected from the real-world PlanetLab test-bed. As discussed in Chapter VI, this is a very limited set of data. There is limited access to the DSL nodes and last mile commercial networks in PlanetLab. The data used in this research are limited to only two nodes. Thus it is hard to say whether these nodes are representative of the the real-world.

3. Limited investigation of prediction: Although, both linear and non-linear prediction methods have be covered in this research, the study is far from being

thorough. The predictors are limited to only linear in parameter models. Also, although the MSE has been found not to be a good criterion for evaluating the predictor performance and ZOC is proposed, it is still not clear how to incorporate the ZOC in a predictor development method.

4. Lack of real-time hardware implementation test: Although the proposed control approach is tested to be effective for real VoIP segments on a realistic test system, the control method is not tested using a real-time hardware test-bed. Additional problems, not investigated in this research, might develop as a result of hardware implementation.

E. Future Work

There are four main areas for future work, as follows:

1. A more scalable method: As discussed above, the current method is still not very scalable for large scale implementation. More research is needed to improve this. Also case studies when the main assumption of this research is violated must be investigated.

2. More real-world data from other sources: In order to verify and be more confident of the voting based predictive path switching method, access to commercial network nodes and DSL nodes, or a better test-bed for the data collection purposes is needed. Otherwise, relying solely on PlanetLab data might bias the outcomes of this research.

3. A better prediction method: The current predictors are not encouraging in their two-step-ahead prediction performance. In most cases, there is a phase shift between the predicted signal and the real signal. This is the biggest problem of

the prediction. The predictors have been optimized using MSE as the objective function. The MSE emphasizes on the accuracy of the predicted signal w.r.t. the original signal. However, in the current research, the sign of the path signal difference is more important than the accuracy of the predicted signal. When the predictors are used in the control strategy, the resultant control performances are not as good as indicated by their MSE during optimization. Some other criteria like ZOC can be used for optimizing the predictors. These criteria are better correlated with the performance of the predictors when used in control. It is possible to optimize the predictors according to these criteria using optimization algorithms like genetic algorithm or simulated annealing. Binary predictors may also be used. However, it is unlikely that new criteria, such as ZOC, will help in removing the time shift in the prediction results. Further studies are needed to come up with better prediction results for better control.

4. A real-time hardware test-bed: The developed control methods need to be implemented and tested on a real-time hardware test-bed for further validation. In order to compare the results, a controlled network environment is needed. In particular, a real-time operating system is needed to cope with the time-varying time-delay of the experiments involved.

REFERENCES

[1] B. Butler. (2006, Mar.) VoIP services ... deep impact. Juniper Research. [Online]. Available: http://www.juniperresearch.com/pdfs/whitepaper_gvoip_hnh.pdf

[2] ——. (2006, Mar.) VoIP ... equipped & ready to go. Juniper Research. [Online]. Available: http://www.juniperresearch.com/pdfs/whitepaper_gvoip_hsa.pdf

[3] B. Goode, "Voice over Internet protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, pp. 1495–1517, Sept. 2002.

[4] Y. Bai and M. R. Ito, "QoS control for video and audio communication in conventional and active networks: Approaches and comparison," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 1, pp. 42–49, First Quarter 2004.

[5] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "The case for resilient overlay networks," in *Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Schloss-Elmau, Germany, May 2001, pp. 152–160.

[6] S. Tao, "Improving the quality of real-time applications through path switching," Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA, 2005.

[7] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71–86, 1997.

[8] B. Tsybakov and N. D. Georganas, "Self-similar processes in communications networks," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1713–1725, Sept. 1998.

[9] M. S. Borella, S. Uludag, G. B. Brewster, and I. Sidhu, "Self-similarity of Internet packet delay," in *Proceedings of IEEE International Conference on Communications*, vol. 1, Montreal, Canada, June 1997, pp. 513–517.

[10] J. le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. New York, NY: Springer-Verlag, 2002.

[11] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti, "Congestion control mechanisms and the best effort service model," *IEEE Network*, vol. 15, no. 3, pp. 16–25, 2001.

[12] S. Low, F. Paganini, and J. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 28–43, 2002.

[13] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, 1989.

[14] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001, pp. 631–640.

[15] D. Harrison, "Edge-to-edge control: a congestion avoidance and service differentiation architecture for the Internet," Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, NY, 2002.

[16] J. Martin, A. Nilsson, and I. Rhee, "Delay-base congestion avoidance for TCP," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 356–369, June 2003.

[17] Y. Xia, D. Harrison, S. Kalyanaraman, K. Ramachandran, and A. Venkatesan, "An accumulation-based congestion control model," in *Proceedings of IEEE International Conference on Communications*, vol. 1, May 11–15, 2003, pp. 657–663.

[18] J. C. R. Bennett and C. Partridge, "Packet reordering is not pathological network behavior," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 789–798, Dec. 1999.

[19] L. Cottrell. (2000, Sept.) Packet reordering. IEPM. [Online]. Available: http://www-iepm.slac.stanford.edu/monitoring/reorder/

[20] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towslay, "Measurement and classification of out-of-Sequence packets in a Tier-1 IP backbone," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002, pp. 213–227.

[21] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering , high speed networks and transport protocol performance," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN'04)*, Chicago, IL, Oct. 2004, pp. 73–78.

[22] J.Bellardo and S.Savage, "Measuring packet reordering," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002, pp. 97–105.

[23] N. M. Piratla, A. P. Jayasumana, and T. Banka, "On reorder density and its

application to characterization of packet reordering," in *Proceedings of the 30th IEEE Conference on Local Computer Networks (LCN 2005)*, Sydney, Australia, Nov. 2005.

[24] N. M. Piratla, A. P. Jayasumana, and A. A. Bare, "A comparative analysis of packet reordering metrics," in *Proceedings of the 1st International Conference on Communication System Software and Middleware (COMSWARE 2006)*, New Delhi, India, Jan. 2006.

[25] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser. (2006, May) Packet reordering metric for IPPM. AT&T Labs, Internet2, and Veriwave. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-ippm-reordering-13.txt

[26] S. Bohacek, J. Hespanha, J. Lee, C. Lim, and K. Obraczka, "A new TCP for persistent packet reordering ," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 369–382, Apr. 2006.

[27] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.

[28] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of the ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 43–56.

[29] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *IEEE Network*, vol. 15, no. 3, pp. 28–37, 2001.

[30] A. Kolarov and G. Ramamurthy, "A control-theoretic approach to the design

of an explicit rate controller for ABR service," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 741–753, Oct. 1999.

[31] M. Sichitiu, P. Bauer, and K. Premaratne, "The effect of uncertain time-variant delays in ATM networks with explicit rate feedback: a control theoretic approach," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 628–637, Aug. 2003.

[32] S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle," *Automatica*, vol. 35, pp. 1921–1935, 1999.

[33] Y. Gu, H. O. Wang, Y. Hong, and L. G. Bushnell, "A predictive congestion control algorithm for high speed communication networks," in *Proceedings of the American Control Conference*, Arlington, VA, June 25–27, 2001, pp. 3779–3780.

[34] V. Misra, W. Gong, and D. F. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proceedings of the ACM SIGCOMM*, 2000, pp. 151–160.

[35] C. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," in *Systems and Control Methods for Communication Networks, Special issue of IEEE Transactions on Automatic Control on*, vol. 47, June 2002, pp. 945–959.

[36] P.-F. Quet, B. Ataslar, A. Iftar, H. Ozbay, S. Kalyanaraman, and T. Kang, "Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays," *Automatica*, vol. 38, no. 6, pp. 917–928, June 2002.

[37] J. Aweya, D. Montuno, Q. Zhang, and L. Orozco-Barbosa, "Congestion control using a multi-step neural predictive technique," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM)*, Sydney, Australia, Nov. 1998, pp. 1705–1714.

[38] R. Atkinson and S. Floyd, "IAB Concerns and Recommendations Regarding Internet Research and Evolution," in *RFC 3869*, Aug. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3869.txt

[39] H. Genel and Y. Erten, "An end-to-end QoS control scheme for video transmission over Internet," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC 2003)*, vol. 2, June 30 – July 3, 2003, pp. 881–886.

[40] W. Jiang and H. Schulzrinne, "Comparisons of FEC and codec robustness on VoIP quality and bandwidth efficiency," in *Proceedings of the 1st IEEE International Conference on Networking (ICN'02)*, Atlanta, GA, 2002.

[41] H. Wu, M. Claypool, and R. Kinicki, "Adjusting forward error correction with temporal scaling for TCP-friendly streaming MPEG," CS Department, Worcester Polytechnic Institute, Tech. Rep. WPI-CS-TR-03-10, Apr. 2003.

[42] X. Chen, C. Wang, D. Xuan, Z. Li, Y. Min, and W. Zhao, "Survey on QoS management of VoIP," in *Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing (ICCNMC 2003)*, Oct. 20–23, 2003, pp. 69–77.

[43] J. Bom, P. Marques, M. Correia, and P. Pinto, "QoS control: an application integrated framework," in *Proceedings of the 1st IEEE International Conference on ATM (ICATM-98)*, June 22–24, 1998, pp. 283–290.

[44] D. Wu, Y. Hou, Y. Zhang, W. Zhu, and H. Chao, "Adaptive QoS control for MPEG-4 video communication over wireless channels," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, Geneva, Switzerland, May 28–31, 2000, pp. 48–51.

[45] P. Bellavista, A. Corradi, and C. Stefanelli, "Application-level QoS control for video-on-demand," *IEEE Internet Computing*, vol. 7, no. 6, pp. 16–24, Nov./Dec. 2003.

[46] S. Lin, Y. Wang, S. Mao, and S. Panwar, "Video transport over ad hoc networks using multiple paths," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, May 2002, pp. I–57 – I–60.

[47] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "A comparative analysis of packet reordering metrics," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC 2003-Fall)*, vol. 5, Oct. 2003, pp. 2961–2965.

[48] T. Okuyama, K. Yasukawa, and K. Yamaoka, "Proposal of multipath routing method focusing on reducing delay jitter," in *Proceedings of the 2005 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 2005)*, Aug. 2005, pp. 296–299.

[49] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," in *Proceedings of the ACM SIGCOMM*, Cambridge, MA, Aug. 1999, pp. 289–299.

[50] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang, "Exploring the performance benefits of end-to-end path switching," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP 2004)*. New York, NY: IEEE, Oct. 2004, pp. 304–315.

[51] S. Tao and R. Guerin, "Application-specific path switching: a case study for streaming video," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*. New York, NY: ACM Press, 2004, pp. 136–143.

[52] P. Morrissey, "Mapping out the best route," *Network Computing*, Dec. 9, 2003. [Online]. Available: http://www.networkcomputing.com/showitem.jhtml?docid=1425f2

[53] J. Kang and B. Nath, "Adaptive QoS control by toggling voice traffic between circuit and packet cellular networks," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM)*. New York: IEEE, 2003, pp. 3498–3503.

[54] "P2P Telephony Explained – For Geeks Only," Skype, 2003. [Online]. Available: http://www.skype.com/products/explained.html

[55] G. Araniti, R. Agarwal, A. Iera, and A. Molinaro, "Multimedia traffic QoS adaptation in UMTS systems through a middleware functionality," in *Proceedings of IEEE International Conference on Communications*, vol. 1, May 11–15, 2003, pp. 17–21.

[56] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Symposium on Operating Systems Principles*, 2001, pp. 131–145.

[57] "Internet bottlenecks: the case for edge delivery services," Akami, 2000. [Online]. Available: http://www.akamai.com/en/resources/pdf/whitepapers/Akamai_Internet_Bottlenecks_Whitepaper.pdf

[58] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: offering

Internet QoS using overlays," *ACM SIGCOMM Computer Communications Review*, vol. 33, no. 1, pp. 11–16, Jan. 2003.

[59] ——, "OverQoS: An overlay based architecture for enhancing Internet QoS," in *Proceeding of the 1st Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004, pp. 71–84.

[60] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, Jan. 2004.

[61] Z.Li and P. Mohapatra, "The impact of topology on overlay routing service," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, Mar. 2004, pp. 408–418.

[62] Z. Ma, H. Shao, and C. Shen, "A new multi-path selection scheme for video streaming on overlay networks," in *Proceedings of IEEE International Conference on Communications*, vol. 3, June 2004, pp. 1330–1334.

[63] X. Gu, K. Nahrstedt, R. Chang, and Z. Shae, "An overlay based QoS-aware voice-over-IP conferencing system," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 3, June 2004, pp. 2111–2114.

[64] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha, "Streaming video over the Internet: Approaches and directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 1, pp. 1–20, Feb. 2001.

[65] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537–549, Aug. 2003.

[66] D. Wu, Y. T. Hou, W. Zhu, H. Lee, T. Chiang, Y. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, Sept. 2000.

[67] "Core technologies of Sony's video conferencing," Sony, July 2003. [Online]. Available: http://www.sony.com.sg/pro/TechnicalPapers/PCS1whitepaper.pdf

[68] Y. Sun, F. Tsou, M. Chen, and Z. Tsai, "A TCP-friendly congestion control scheme for real-time packet video using prediction," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, 1999, pp. 1818–1822.

[69] S. Wee, W. Tan, J. Apostolopoulos, and M. Etoh, "Optimized video streaming for networks with varying delay," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 2, Lausanne, Switzerland, Aug. 2002, pp. 89–92.

[70] S. C. Liew and D. C. Tse, "A control-theoretic approach to adapting VBR compressed video for transport over a CBR communications channel," *IEEE/ACM Transactions on Networking*, vol. 6, no. 1, pp. 42–55, Feb. 1998.

[71] S. Chakrabarti and R. Wang, "Adaptive control for packet video," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, 1994, pp. 56–62.

[72] "Packet-based multimedia communications systems," in *ITU-T Recommendation H.323*. Geneva, Switzerland: ITU, 2003.

[73] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," in *RFC 3261*, June 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[74] "Methods for subjective determination of transmission quality," in *ITU-T Recommendation P.800*. Geneva, Switzerland: ITU, 1996.

[75] "The E-model, a computational model for use in transmission planning," in *ITU-T Recommendation G.107*. Geneva, Switzerland: ITU, 2000.

[76] A. W. Rix, M. P. Hollier, A. P. Hekstra, and J. G. Beerends, "Perceptual evaluation of speech quality (PESQ): The new ITU standard for end-to-end speech quality assessment part I - time-delay compensation," *Journal of the Audio Engineering Society*, vol. 50, no. 10, pp. 755–764, Oct. 2002.

[77] C. Shim, L. Xie, B. Zhang, and C. Sloane, "How delay and packet loss impact voice quality in VoIP," Qovia Inc., Frederick, MD, Tech. Rep., Dec. 2003.

[78] H. Furuya, S. Nomoto, H. Yamada, N. Fukumoto, and F. Sugaya, "Toward QoS management of VoIP: experimental investigation of the relations between IP network performances and VoIP speech quality," *IEICE Transactions on Communications*, vol. E87-B, no. 6, pp. 1610–1622, June 2004.

[79] E. E. Zurek, J. Leffew, and W. A. Moreno, "Objective evaluation of voice clarity measurements for VoIP compression algorithms," in *Proceedings of the 4th IEEE International Caracas Conference on Devices, Circuits and Systems*, Apr. 17–19, 2002, pp. T033–1 – T033–6.

[80] A. W. Rix, "Comparison between subjective listening quality and P.862 PESQ score," in *Proceedings of Online Workshop Measurement of Speech and Audio*

Quality in Networks (MESAQIN), Prague, Czech Republic, May 2003, pp. 17–25.

[81] P. Galiotos, T. Dagiuklas, and D. Arkadianos, "QoS management for an enhanced VoIP platform R-factor and network load estimation," in *Proceedings of the 5th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC02)*, July 2002, pp. 305–314.

[82] L. Ding and R. Goubran, "Speech quality prediction in VoIP using the extended E-model," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 7, Dec. 2003, pp. 3974–3978.

[83] L. Sun, "Speech quality prediction for voice over internet protocol networks," Ph.D. dissertation, University of Plymouth, Plymouth, U.K., 2004.

[84] A. Takahashi, H. Yoshino, and N. Kitawaki, "Perceptual QoS Assessment Technologies for VoIP," *IEEE Communications Magazine*, vol. 42, no. 7, pp. 28–34, July 2004.

[85] H. Melvin and L. Murphy, "Time synchronization for VoIP quality of service," *IEEE Internet Computing*, vol. 6, no. 3, pp. 57–63, May 2002.

[86] D. L. Mills, "Network Time Protocol (Version 3)," in *RFC 1305*, Mar. 1992. [Online]. Available: http://www.ietf.org/rfc/rfc1305.txt

[87] S. Johannessen, "Time synchronization in a local area network," *IEEE Control Systems Magazine*, vol. 24, no. 2, pp. 61–69, Apr. 2004.

[88] V. Paxson, "On calibrating measurements of packet transit times," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, 1998, pp. 11–21.

[89] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, Mar. 1999, pp. 227–234.

[90] L. Zhang, Z. Liu, and C. Xia, "Clock synchronization algorithms for network measurements," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, June 2002, pp. 160–169.

[91] A. Bletsas, "Evaluation of Kalman filtering for network time keeping," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, Mar. 2003, pp. 289–296.

[92] P. S. A. Mohamed, "Automatic Evaluation of Real-Time Multimedia Quality: a Neural Network Approach," Ph.D. dissertation, Institut de recherche en informatique et systmes alatoires (IRISA), IRISA / INRIA Rennes, Campus universitaire de Beaulieu, France, 2003.

[93] J. Valin, *The Speex Codec manual*, The Speex project, Mar. 2003. [Online]. Available: http://www.speex.org

[94] H. Oouch, T. Takenage, H. Sugawara, and M. Masugi, "Study on appropriate voice data length of IP packets for VoIP network adjustment," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 2, Nov. 2002, pp. 1618–1622.

[95] G. Scheets, R. Singh, and M. Parperis, "Analyzing end-to-end delivery delay in pure VoIP networks," in *Proceedings of the 45th Midwest Symposium on Circuits and Systems (MWSCAS-2002)*, vol. 2, Aug. 2002, pp. II–517–II–520.

[96] G. Scheetes, M. Parperis, and R. Singh, "Voice over the Internet: A tutorial discussing problems and solutions associated with alternative transport," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 22–31, Second quarter 2004.

[97] A. Bilhaj and K. Mase, "Endpoint admission control enhanced systems for VoIP networks," *IEICE Transactions on Communications*, vol. E87-B, no. 4, pp. 948–957, Apr. 2004.

[98] J. Bolot and A. Vega-Garcia, "Control mechanisms for packet audio in the internet," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 1996, pp. 232–239.

[99] S. Floyd and J. Kempf, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet," in *RFC 3714*, Mar. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3714.txt

[100] K. Homayounfar, "Rate adaptive speech coding for universal multimedia access," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 30–39, Mar. 2003.

[101] Z. Qiao, L. Sun, N. Heilemann, and E. Ifeachor, "A new method for VoIP quality of service control use combined adaptive sender rate and priority marking," in *Proceedings of IEEE International Conference on Communications*, vol. 3, June 2004, pp. 1473–1477.

[102] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang, "Improving VoIP quality through path switching," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 4, Mar. 2005, pp. 2268–2278.

[103] L. Sun and E. C. Ifeachor, "Prediction of perceived conversational speech quality and effects of playout buffer algorithms," in *Proceedings of IEEE International Conference on Communications*, vol. 1, May 2003, pp. 1–6.

[104] M. Ranganathan and L. Kilmanrtin, "Neural and fuzzy computation techniques for playout delay adaptation in VoIP networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1174–1194, Sept. 2005.

[105] R. Prasad, C. Dovrolis, M. Murry, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, pp. 27–35, Nov./Dec. 2003.

[106] A. Shriram, M. Murray, H. Young, N. Brownless, A. Broido, M. Fomenkov, and K. Claffy, "Comparison of public end-to-end bandwidth estimation tools on high-speed links," in *Proceedings of the Passive and Active Measurement (PAM) Workshop*, vol. 3431, Boston, MA, Mar. 31, 2005, pp. 306–320.

[107] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, Aug. 2001.

[108] S. Floyd and E. Kohler, "Internet research needs better models," in *Proceedings of Workshop on Hot Topics in Networks*, Oct. 28–29, 2002.

[109] "Welcome to the PlanetLab website," PlanetLab, July 2005. [Online]. Available: http://www.planet-lab.org/

[110] N. Spring, L. Peterson, A. Bavier, and V. Pai, "Using PlanetLab for network research: myths, realities, and best practices," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 17–24, Jan. 2006.

[111] "NIST Net Home Page," National Institute of Standards and Technology, July 2005. [Online]. Available: http://snad.ncsl.nist.gov/itg/nistnet/

[112] S. McCanne and S. Floyd. (2002) Network Simulator 2 (ns-2). Information Science Institute, University of Southern California. Los Angeles, CA. [Online]. Available: http://www.isi.edu/nsnam/ns/

[113] B. Liu, D. R. Figueiredo, Y. Guo, J. F. Kurose, and D. F. Towsley, "A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001, pp. 1244–1253.

[114] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale IP networks," in *Proceedings of the 2003 ACM SIGMETRICS International Conference*, San Diego, CA, June 10–14, 2003, pp. 91–101.

[115] A. G. Parlos and D. Ye, "A transport model for flow processes characterized by time-varying time delays," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, Maui, HI, Dec. 2003, pp. 2618–2619.

[116] A. Konstantinou, "Adaptive flow control techniques for media applications in best effort networks using fluid models ," M.S. thesis, Texas A&M University, College Station, TX, 2004.

[117] H. Ohsaki, M. Morita, and M. Murata, "On modeling round-trip dynamics of the Internet using system identification," *IEICE Transactions on Communications*, vol. E85-B, no. 1, pp. 1–9, Jan. 2002.

[118] A. Bremler-Barr, E. Cohen, H. Kaplan, and Y. Mansour, "Predicting and bypassing end-to-end Internet service degradations," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 961–978, June 2003.

[119] S. Tao and R. Guerin, "On-line estimation of internet path performance: An application perspective," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 2004, pp. 1774–1785.

[120] M. Hasegawa, G. Wu, and M. Mizuno, "Applications of nonlinear prediction methods to the internet traffic," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, Sydney, Australia, May 2001, pp. 169–172.

[121] R. Boné and M. Crucianu, "Multi-step-ahead prediction with neural networks: a review," in *Approches Connexionnistes en Sciences Économiques et en Gestion*, Boulogne sur Mer, France, Nov. 2002, pp. 97–106.

[122] Q. Wang, D. Tan, X. Ning, and Y. Wang, "The control oriented QoS: analysis and prediction," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2001, pp. 1897–1902.

[123] M. Kommaraju, "Predictor development for controlling real-time applications over the internet," M.S. thesis, Texas A&M University, College Station, TX, 2005.

[124] K. Shah, S. Bohacek, and E. Jonckheere, "On predictability of data network traffic," in *Proceedings of the American Control Conference*, Denver, CO, June 2003, pp. 1619–1624.

[125] A. G. Parlos, "Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, May 12–17, 2002, pp. 2460–2465.

[126] S. Doddi, "Emperical modeling of end-to-end delay dynamics in best-effort networks," M.S. thesis, Texas A&M University, College Station, TX, 2004.

[127] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Chapel Hill, NC, June 26–28, 2000, pp. 271–280. [Online]. Available: http://www.nossdav.org/2000/papers/27.pdf

[128] H. R. Mehrvar and M. R. Soleymani, "Packet loss rate prediction using a universal indicator of traffic," in *Proceedings of IEEE International Conference on Communications*, vol. 3, Helsinki, Finland, June 2001, pp. 647–653.

[129] L. Roychoudhuri and E. S. Al-Shaer, "Real-time analysis of delay variation for packet loss prediction." in *Proceedings of the 7th IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS 2004)*, San Diego, CA, Oct. 2004, pp. 213–227. [Online]. Available: http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=3271&spage=213

[130] V. Khariwal, "Adaptive control of media applications in best effort networks," M.S. thesis, Texas A&M University, College Station, TX, 2004.

[131] P. Bauer, M. Sichitiu, and K. Premaratne, "Closing the loop through communication networks: The case of an integrator plant and multiple controllers," in

Proceedings of the IEEE Conference on Decision and Control, Dec. 7–10, 1999, pp. 2180–2185.

[132] S. Niculescu, M. Fu, and H. Li, "Delay-dependent closed-loop stability of linear systems with input delay: an LMI approach," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, San Diego, CA, Dec. 1997, pp. 1623–1628.

[133] X. Nian, "Stability of linear systems with time-varying delays: an Lyapunov functional approach," in *Proceedings of the American Control Conference*, vol. 1, June 4–6, 2003, pp. 883–884.

[134] M. M. Ekanayake, "Stability of discrete-time systems with time-varying delays," in *Proceedings of the American Control Conference*, vol. 5, June 25–27, 2001, pp. 3914–3919.

[135] E. F. Camacho and C. Bordons, *Model Predictive Control*. New York, NY: Springer-Verlag, 1999.

[136] J. A. Rossiter, *Model-based Predictive Control : A Practical Approach*. Boca Raton, FL: CRC Press, 2003.

[137] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[138] "Objective quality measurement of telephone-band (300-3400Hz) speech codecs," in *ITU-T Recommendation P.861*. Geneva, Switzerland: ITU, 1998.

[139] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) – a new method for speech quality assessment of telephone networks and codecs," in *Proceedings of the IEEE International*

Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, May 7–11, 2001, pp. 749–752.

[140] "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," in *ITU-T Recommendation P.862*. Geneva, Switzerland: ITU, 2001.

[141] A. Estepa, R. Estepa, and J. M. Vzmediano, "On the suitability of the E-model to VoIP networks," in *Proceedings of the 7th IEEE International Symposium on Computers and Communication (ISCC 2002)*, July 2002, pp. 511–516.

[142] L. Ding and R. A. Goubran, "Assessment of effects of packet loss on speech quality in VoIP," in *Proceedings of the 2nd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications (HAVE 2003)*, Sept. 20–21, 2003, pp. 49–54.

[143] B. Duysburgh, S. Vanhastel, B. Vreese, C. Petrisor, and P. Demeester, "On the influence of best-effort network conditions on the perceived speech quality of VoIP connections," in *Proceedings of the 10th International Conference on Computer Communications and Networks (ICCCN'01)*, Scottsdale, AZ, 2001, pp. 334–339.

[144] S. Kay, *Fundamentals of Statistical Processing, Volume I: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[145] A. P. Black, J. Huang, R. Koster, J. Walpole, and C. Pu, "Infopipes: An abstraction for multimedia streaming," Department of Computer Science and Engineering, Tech. Rep. CSE 02-001, OGI School of Science and Engineering, Portland, OR, Jan. 2002.

[146] P.-F. Quet, S. Ramakrishnan, H. Ozbay, and S. Kalyanaraman, "On the H-infinity controller design for congestion control in communication networks with a capacity predictor," in *Proceedings of the IEEE Conference on Decision and Control*, Dec. 2001, pp. 598–603.

[147] B. Ataslar, P.-F. Quet, A. Iftar, H. Ozbay, and T. Kang, "Robust rate-based flow controllers for high-speed networks: The case of uncertain time-varying multiple time-delays," in *Proceedings of the American Control Conference*, Chicago, IL, June 2000, pp. 2804–2808.

[148] "One-way transmission time," in *ITU-T Recommendation G.114*. Geneva, Switzerland: ITU, May 2003.

[149] "Understanding Delay in Packet Voice Networks," White Papers, Cisco, San Jose, CA, Feb. 2006. [Online]. Available: http://www.cisco.com/warp/public/788/voip/delay-details.html

[150] Wikipedia, "Normal distribution — Wikipedia, the free encyclopedia," July 15, 2006. [Online]. Available: http://en.wikipedia.org/wiki/Gaussian_distribution

[151] ——, "Pareto distribution — Wikipedia, the free encyclopedia," July 15, 2006. [Online]. Available: http://en.wikipedia.org/wiki/Pareto_distribution

[152] J. Strater and S. Nikola, "Engineering CMTS and HFC for VoIP with Capital and Operating Expenses in Mind," in *SCTE and Communication Technology's Advanced VoIP Development Symposium 2004*, Sept. 2004. [Online]. Available: http://broadband.motorola.com/catalog/product_documents/CMTSDesign_Considerations_for_VoIP.pdf

[153] "IP Telephony design Guide," Alcatel, Apr. 2003. [Online].

Available: http://www.alcatel.com/enterprise/en/resource_library/pdf/wp/wp_IPT_Design-Guide.pdf

[154] Y. Shukla, "Prediction of end-to-end single flow characteristics in best-effort networks," M.S. thesis, Texas A&M University, College Station, TX, 2005.

[155] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.

[156] "SEMATECH e-Handbook of Statistical Methods," NIST, May 2006. [Online]. Available: http://www.itl.nist.gov/div898/handbook/

[157] G. E. Box and G. M. Jenkins, *Time Series Analysis : Forecasting and Control*. San Francisco, CA: Holden-Day, 1976.

[158] T. Soderstrom and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[159] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[160] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[161] S. Chen and S. Billings, "Representations of nonlinear systems: the NARMAX model," *International Journal of Control*, vol. 49, pp. 1013–1032, 1989.

[162] S. Chen, S. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, pp. 1873–1896, 1989.

[163] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[164] H. Demuth, M. Beale, and M. Hagan, *Neural Network Toolbox User's Guide*, 5th ed. Natick, MA: The MathWorks, 1992–2006.

[165] R. Jordan, *The Fires of Heaven* [sound recording]. New York, NY: Audio Renaissnce, 1997.

APPENDIX A

ADDITIONAL VOICE QUALITY CONTROL RESULTS

Table LIII presents the resulting average CLR and PESQ-MOS of the different controls for the 8th trace-file of path pair AB. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A method (increments of 0.11 and 0.19, respectively) and Path B method (increments of 1.04 and 1.12, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.31 and 1.24 over Path A method and Path B method, respectively) and Ideal control (increments of 0.50 and 1.43 over Path A method and Path B method, respectively).

Table LIII. Average CLR and PESQ-MOS of #8 trace-file path pair AB.

	Path A	Path B	SP_{CLR}	V7[1]	OSV7[2]	Ideal
Avg.[3] CLR (%)	14.64	18.24	5.49	4.84	3.70	2.83
SD[4] CLR (%)	16.99	5.44	3.78	3.72	3.84	4.07
Avg. PESQ-MOS	3.63	2.70	3.74	3.82	3.94	4.13
SD PESQ-MOS	1.28	1.20	1.01	0.96	0.88	0.84

[1] Voting based predictive path switching control.

[2] Voting based one step ahead predictive path switching control.

[3] Average.

[4] Standard deviation.

Table LIV shows the voice quality distribution of each control for the 8th trace-file of path pair AB. It shows that SP_{CLR} control is better than both Path A method (23.51% bad quality segments vs 25.95% bad quality segments), and Path B method

(23.51% bad quality segments vs 61.62% bad quality segments), V7 control is better than both Path A method (19.73% bad segments vs 25.95% bad segments) and Path B method (19.73% bad segments vs 61.62% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (13.51% bad segments). The Ideal control gives the best result (10.54% bad segments).

Table LIV. Voice quality distribution of #8 trace-file path pair AB.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A	25.95	4.05	70.00
B	61.62	8.65	29.73
SP_{CLR}	23.51	8.65	67.84
V7	19.73	8.11	72.16
OSV7	13.51	7.84	78.65
Ideal	10.54	4.05	85.41

Table LV gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 8^{th} trace-file of path pair AB. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path B method are positive, they improve the voice quality compared to Path B method. The percentage improvements of V7 control over Path A method are positive, it improves the voice quality compared to Path A method. But the percentage improvement of SP_{CLR} control over Path A method is negative, it is doing worse than Path A method. The OSV7 control results indicates that if there is no feedback delay, then the predictive path switching control is always

better than no switching methods. The ideal case predictive path switching control gives the best results.

Table LV. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #8 trace-file path pair AB.

	A	B	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	25.95	61.62	23.51	19.73	13.51	10.54
Improvement over path A (%)			9.38	23.96	47.92	59.38
Improvement over path B (%)			61.84	67.98	78.07	82.89
Improvement over SP_{CLR} (%)				16.09	42.53	55.17
PESQ-MOS < 3.5 (%)	30.00	70.27	32.16	27.84	21.35	14.59
Improvement over path A (%)			-7.21	7.21	28.83	51.35
Improvement over path B (%)			54.23	60.38	69.62	79.23
Improvement over SP_{CLR} (%)				13.45	33.61	54.62

Table LVI is the comparison of the predictive path switching control and the no switching methods for the 8^{th} trace-file of path pair AB. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path B method, there are more improvements from SP_{CLR} control and V7 control than degradations. Compared to Path A method, the V7 control is still better.

Table LVII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 8^{th} trace-file of path pair AB. A zoom-in version is plotted in Fig. 73. From the distribution and the plots, it can be seen that the predictive path switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the

Table LVI. Comparison of predictive path switching control and no switching methods for #8 trace-file path pair AB.

	Comparing to path A (%)			Comparing to path B (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	20.27	58.92	20.81	62.70	24.86	12.43
V7	22.16	64.32	13.51	67.03	18.92	14.05
OSV7	29.46	62.97	7.57	71.35	18.65	10.00
Ideal	37.57	61.62	0.81	73.78	25.68	0.54

PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LVII. PESQ-MOS distributions calculated over every 0.5 MOS point for #8 trace-file path pair AB.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A	14.59	2.97	3.78	4.59	4.05	9.46	60.54
Path B	21.08	11.35	13.24	15.95	8.65	9.19	20.54
SP_{CLR}	4.86	3.24	6.22	9.19	8.65	12.43	55.41
V7	4.05	2.97	5.41	7.30	8.11	14.86	57.30
OSV7	2.97	2.70	3.51	4.32	7.84	15.95	62.70
Ideal	2.70	2.43	2.43	2.97	4.05	6.22	79.19

The cumulative PESQ-MOS distribution is plotted in Fig. 74. Below MOS level of 3, the plots of predictive path switching controls are lower (better) than the no switching methods. At MOS level of 3, plot of SP_{CLR} control matches that of Path A method, and above that MOS level the plot of SP_{CLR} control is higher (worse) than

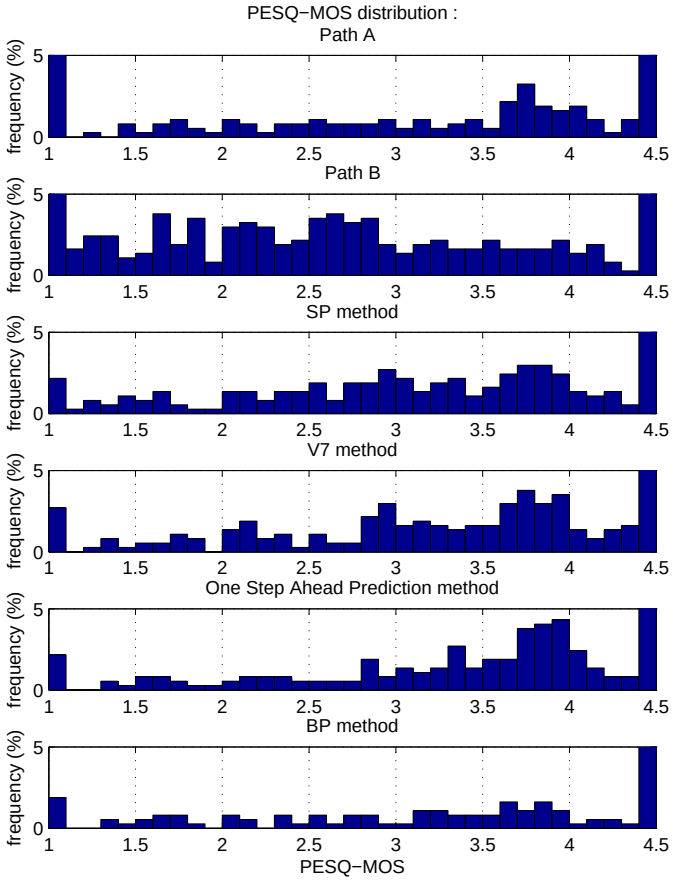

Fig. 73. Zoom-in PESQ-MOS distribution of #8 trace-file path pair AB.

Path A method. The plot of V7 control matches with Path A method after MOS level of 3.5, which indicates V7 control improves the bad quality segments (below MOS level of 2) to the fair quality range (around MOS level of 3), but have difficulty in improving them into the good quality level (above MOS level of 3.5). The plot of OSV7 control is lower (better) than SP_{CLR} control and V7 control. The plot of Ideal control is the lowest (best).

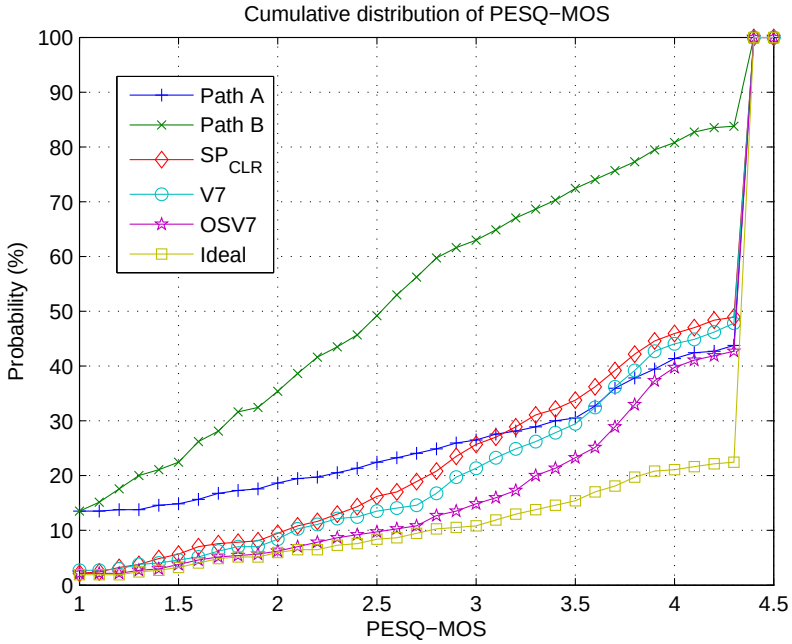

Fig. 74. Cumulative PESQ-MOS distribution of #8 trace-file path pair AB.

Table LVIII presents the resulting average CLR and PESQ-MOS of the different controls for the 4th trace-file of path pair AC. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A method (increments of 0.42 and 0.49, respectively) and Path C method (increments of 0.25 and 0.32, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.66 and 0.49 over Path A method and Path C method, respectively) and Ideal control (increments of 0.86 and 0.69 over Path A method and Path C method, respectively).

Table LVIII. Average CLR and PESQ-MOS of #4 trace-file path pair AC.

	Path A	Path C	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	9.22	7.32	3.21	2.92	1.56	0.65
SD CLR (%)	4.09	4.20	1.40	1.58	1.13	0.80
Avg. PESQ-MOS	3.54	3.71	3.96	4.03	4.20	4.40
SD PESQ-MOS	1.11	1.14	0.74	0.68	0.54	0.39

Table LIX shows the voice quality distribution of each control for the 4th trace-file of path pair AC. It shows that SP_{CLR} control is better than both Path A method (13.78% bad quality segments vs 26.49% bad quality segments), and Path C method (13.78% bad quality segments vs 22.97% bad quality segments), V7 control is better than both Path A method (10.00% bad segments vs 26.49% bad segments) and Path C method (10.00% bad segments vs 22.97% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (4.05% bad segments). The Ideal control gives the best result (2.97% bad segments).

Table LIX. Voice quality distribution of #4 trace-file path pair AC.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A	26.49	11.08	62.43
C	22.97	7.30	69.73
SP_{CLR}	13.78	13.24	72.97
V7	10.00	11.89	78.11
OSV7	4.05	5.95	90.00
Ideal	2.97	1.08	95.95

Table LX gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 4^{th} trace-file of path pair AC. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A method and Path C method are positive, they improve the voice quality compared to Path A method and Path B method. The OSV7 control results are even better than the results of SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXI is the comparison of the predictive path switching control and the no switching methods for the 4^{th} trace-file of path pair AC. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A method and Path C method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 4^{th} trace-file of path pair AC. A zoom-in version is plotted in Fig. 75. From the distribution and the plots, it can be seen that the predictive path switching

Table LX. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #4 trace-file path pair AC.

	A	C	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	26.49	22.97	13.78	10.00	4.05	2.97
Improvement over path A (%)			47.96	62.24	84.69	88.78
Improvement over path C (%)			40.00	56.47	82.35	87.06
Improvement over SP_{CLR} (%)				27.45	70.59	78.43
PESQ-MOS < 3.5 (%)	37.57	30.27	27.03	21.89	10.00	4.05
Improvement over path A (%)			28.06	41.73	73.38	89.21
Improvement over path C (%)			10.71	27.68	66.96	86.61
Improvement over SP_{CLR} (%)				19.00	63.00	85.00

Table LXI. Comparison of predictive path switching control and no switching methods for #4 trace-file path pair AC.

	Comparing to path A (%)			Comparing to path C (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	41.35	39.73	18.92	27.03	54.59	18.38
V7	47.30	28.92	23.78	24.32	68.11	7.57
OSV7	51.08	33.51	15.41	33.78	60.54	5.68
Ideal	56.22	43.78	0.00	41.62	58.38	0.00

controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXII. PESQ-MOS distributions calculated over every 0.5 MOS point for #4 trace-file path pair AC.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A	8.65	5.14	4.59	8.11	11.08	14.86	47.57
Path C	9.19	4.05	5.41	4.32	7.30	8.92	60.81
SP_{CLR}	0.54	1.35	3.51	8.38	13.24	10.54	62.43
V7	0.54	0.81	2.16	6.49	11.89	13.51	64.59
OSV7	0.54	0.54	0.81	2.16	5.95	15.95	74.05
Ideal	0.54	0.00	0.54	1.89	1.08	2.43	93.51

The cumulative PESQ-MOS distribution is plotted in Fig. 76. Below MOS level of 4, the plots of predictive path switching controls are lower (better) than the no switching methods. At MOS level of 4, plot of SP_{CLR} control matches that of Path B method. This indicates the predictive path switching controls improve the speech segments qualities at both bad and fair MOS level here. But above MOS level of 3, the plot of SP_{CLR} control is close to that of Path C method, and above MOS level of 3.5 the plot of V7 control is also close to that of Path C method, which indicates the predictive path switching controls give more improvement to the bad quality segments (below MOS level of 2) than to the fair quality segments (around MOS level of 3).

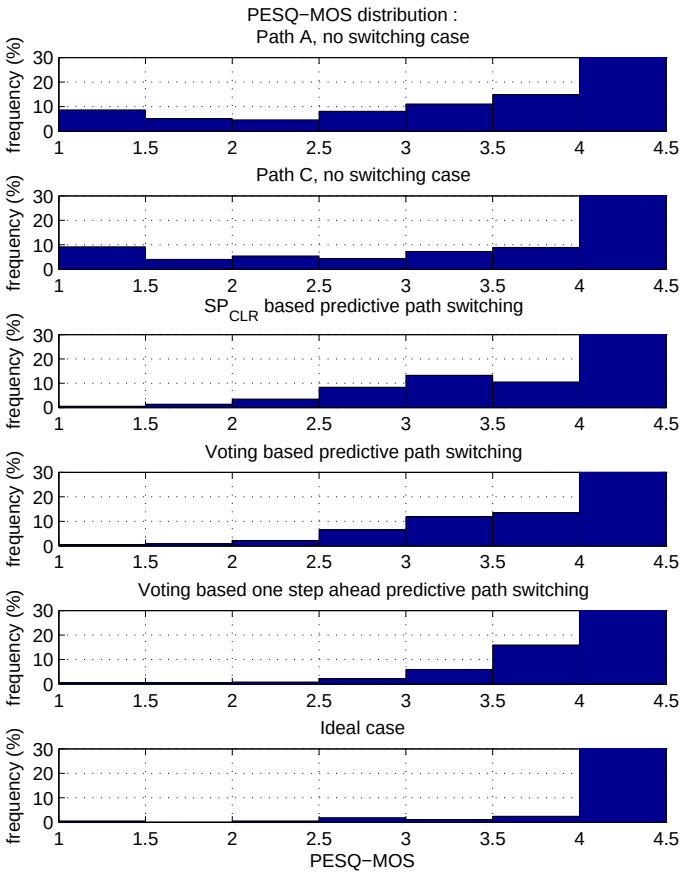

Fig. 75. Zoom-in PESQ-MOS distribution of #4 trace-file path pair AC.

The plot of OSV7 control is lower (better) than SP_{CLR} control and V7 control. The plot of Ideal control is the lowest (best).

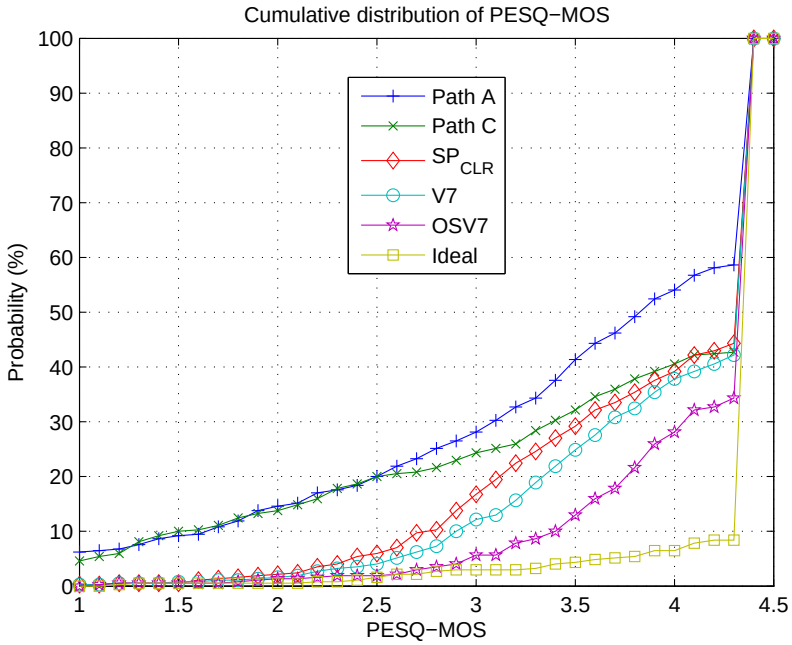

Fig. 76. Cumulative PESQ-MOS distribution of #4 trace-file path pair AC.

Table LXIII presents the resulting average CLR and PESQ-MOS of the different controls for the 8th trace-file of path pair AC. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A method (increments of 0.44 and 0.47, respectively) and Path C method (increments of 0.15 and 0.18, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.60 and 0.31 over Path A method and Path C method, respectively) and Ideal control (increments of 0.73 and 0.44 over Path A method and Path C method, respectively).

Table LXIII. Average CLR and PESQ-MOS of #8 trace-file path pair AC.

	Path A	Path C	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	14.64	6.08	3.69	3.47	2.66	2.15
SD CLR (%)	16.99	4.39	4.10	3.85	3.86	3.91
Avg. PESQ-MOS	3.63	3.92	4.07	4.10	4.23	4.36
SD PESQ-MOS	1.28	0.99	0.77	0.72	0.63	0.59

Table LXIV shows the voice quality distribution of each control for the 8th trace-file of path pair AC. It shows that SP_{CLR} control is better than both Path A method (9.73% bad quality segments vs 25.95% bad quality segments), and Path C method (9.73% bad quality segments vs 14.86% bad quality segments), V7 control is better than both Path A method (7.30% bad segments vs 25.95% bad segments) and Path C method (7.30% bad segments vs 14.86% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (4.32% bad segments). The Ideal control gives the best result (2.97% bad segments).

Table LXIV. Voice quality distribution of #8 trace-file path pair AC.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A	25.95	4.05	70.00
C	14.86	7.57	77.57
SP_{CLR}	9.73	7.30	82.97
V7	7.30	7.84	84.86
OSV7	4.32	4.59	91.08
Ideal	2.97	0.81	96.22

Table LXV gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 8^{th} trace-file of path pair AC. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A method and Path C method are positive, they improve the voice quality compared to Path A method and Path B method. The OSV7 control results are even better than the results of SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXVI is the comparison of the predictive path switching control and the no switching methods for the 8^{th} trace-file of path pair AC. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A method and Path C method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXVII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 8^{th} trace-file of path pair AC. A zoom-in version is plotted in Fig. 77. From the distribution and the plots, it can be seen that the predictive path

Table LXV. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #8 trace-file path pair AC.

	A	C	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	25.95	14.86	9.73	7.30	4.32	2.97
Improvement over path A (%)			62.50	71.88	83.33	88.54
Improvement over path C (%)			34.55	50.91	70.91	80.00
Improvement over SP_{CLR} (%)				25.00	55.56	69.44
PESQ-MOS < 3.5 (%)	30.00	22.16	17.03	15.14	8.92	3.78
Improvement over path A (%)			43.24	49.55	70.27	87.39
Improvement over path C (%)			23.17	31.71	59.76	82.93
Improvement over SP_{CLR} (%)				11.11	47.62	77.78

Table LXVI. Comparison of predictive path switching control and no switching methods for #8 trace-file path pair AC.

	Comparing to path A (%)			Comparing to path C (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	31.89	53.78	14.32	18.38	67.84	13.78
V7	37.03	43.78	19.19	18.11	74.86	7.03
OSV7	39.19	49.73	11.08	25.95	71.35	2.70
Ideal	41.89	58.11	0.00	30.81	68.65	0.54

switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXVII. PESQ-MOS distributions calculated over every 0.5 MOS point for #8 trace-file path pair AC.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A	14.59	2.97	3.78	4.59	4.05	9.46	60.54
Path C	5.41	2.43	2.43	4.59	7.57	10.81	66.76
SP_{CLR}	1.89	0.81	2.16	4.86	7.30	12.70	70.27
V7	1.62	1.08	1.08	3.51	7.84	13.78	71.08
OSV7	1.89	0.54	0.00	1.89	4.59	11.35	79.73
Ideal	2.16	0.54	0.00	0.27	0.81	2.97	93.24

The cumulative PESQ-MOS distribution is plotted in Fig. 78. Here, the plots of predictive path switching controls are lower (better) than the no switching methods. The distance is quite large at both MOS level of 3, and MOS level of 3.5, which indicates the predictive path switching controls give improvement to both the bad quality segments (below MOS level of 2) and the fair quality segments (around MOS level of 3). The plot of OSV7 control is lower (better) than SP_{CLR} control and V7 control. The plot of Ideal control is the lowest (best).

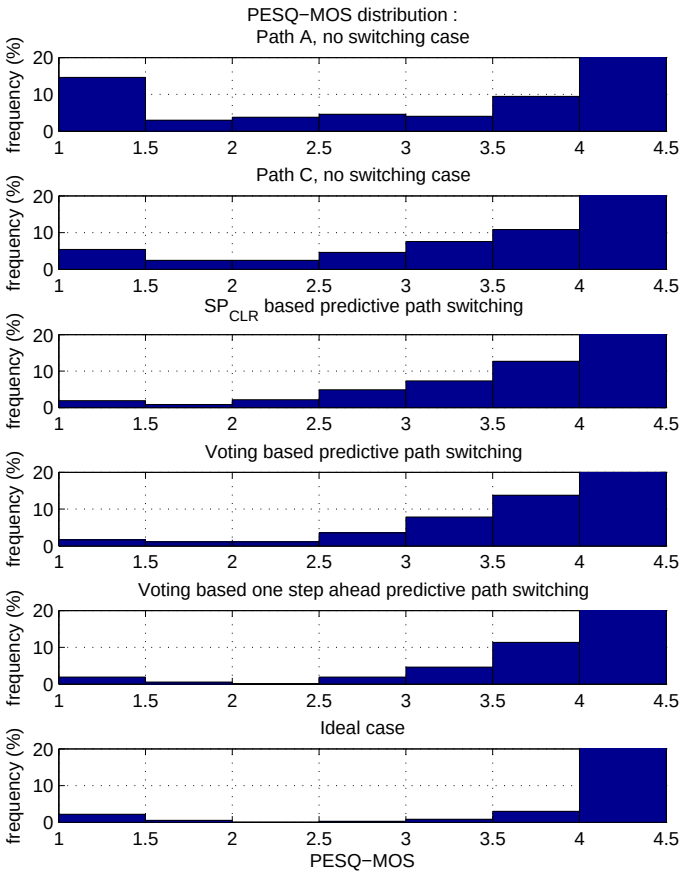

Fig. 77. Zoom-in PESQ-MOS distribution of #8 trace-file path pair AC.

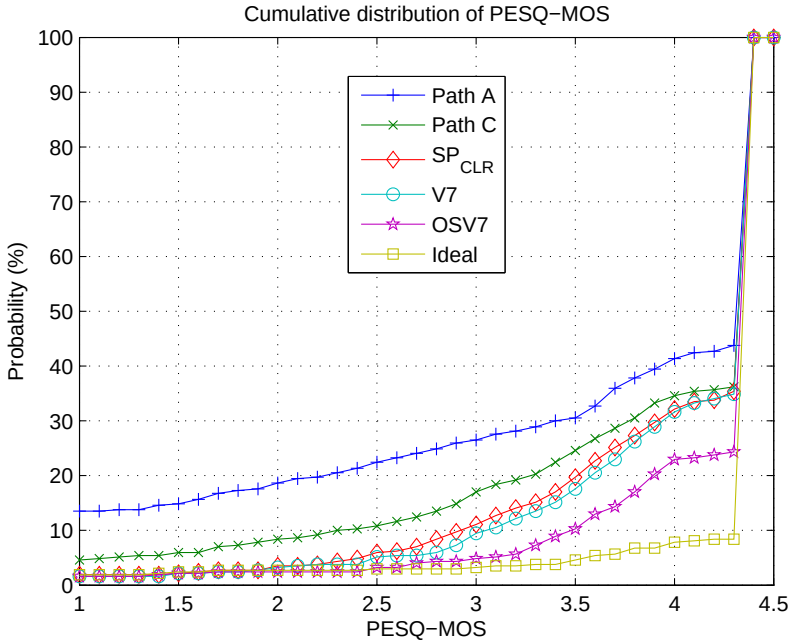

Fig. 78. Cumulative PESQ-MOS distribution of #8 trace-file path pair AC.

Table LXVIII presents the resulting average CLR and PESQ-MOS of the different controls for the 4th trace-file of path pair $A_{new}B_{new}$. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A_{new} method (increments of 0.20 and 0.16, respectively) and Path B_{new} method (increments of 0.73 and 0.69, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.30 and 0.83 over Path A_{new} method and Path B_{new} method, respectively) and Ideal control (increments of 0.47 and 1.00 over Path A_{new} method and Path B_{new} method, respectively).

Table LXVIII. Average CLR and PESQ-MOS of #4 trace-file path pair $A_{new}B_{new}$.

	Path A_{new}	Path B_{new}	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	5.61	13.58	2.09	2.16	1.27	0.51
SD CLR (%)	2.13	7.42	1.25	1.09	0.78	0.54
Avg. PESQ-MOS	3.94	3.41	4.14	4.10	4.24	4.41
SD PESQ-MOS	1.02	1.28	0.67	0.67	0.50	0.35

Table LXIX shows the voice quality distribution of each control for the 4th trace-file of path pair $A_{new}B_{new}$. It shows that SP_{CLR} control is better than both Path A_{new} method (7.57% bad quality segments vs 14.05% bad quality segments), and Path B_{new} method (7.57% bad quality segments vs 32.43% bad quality segments), V7 control is better than both Path A_{new} method (7.30% bad segments vs 14.05% bad segmetns) and Path B_{new} method (7.30% bad segments vs 32.43% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (2.97% bad segments). The Ideal control gives the best result (1.08% bad segments).

Table LXIX. Voice quality distribution of #4 trace-file path pair $A_{new}B_{new}$.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A_{new}	14.05	4.32	81.62
B_{new}	32.43	8.65	58.92
SP_{CLR}	7.57	8.65	83.78
V7	7.30	8.38	84.32
OSV7	2.97	4.86	92.16
Ideal	1.08	2.16	96.76

Table LXX gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 4^{th} trace-file of path pair $A_{new}B_{new}$. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A_{new} method and Path B_{new} method are positive, they improve the voice quality compared to Path A_{new} method and Path B_{new} method. The OSV7 control results are better than SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXXI is the comparison of the predictive path switching control and the no switching methods for the 4^{th} trace-file of path pair $A_{new}B_{new}$. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A_{new} method and Path B_{new} method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXXII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 4^{th} trace-file of path pair $A_{new}B_{new}$. A zoom-in version is plotted

Table LXX. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #4 trace-file path pair $A_{new}B_{new}$.

	A	B	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	14.05	32.43	7.57	7.30	2.97	1.08
Improvement over path A (%)			46.15	48.08	78.85	92.31
Improvement over path B (%)			76.67	77.50	90.83	96.67
Improvement over SP_{CLR} (%)				3.57	60.71	85.71
PESQ-MOS < 3.5 (%)	18.38	41.08	16.22	15.68	7.84	2.97
Improvement over path A (%)			11.76	14.71	57.35	83.82
Improvement over path B (%)			60.53	61.84	80.92	92.76
Improvement over SP_{CLR} (%)				3.33	51.67	81.67

Table LXXI. Comparison of predictive path switching control and no switching methods for #4 trace-file path pair $A_{new}B_{new}$.

	Comparing to path A_{new} (%)			Comparing to path B_{new} (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	20.54	64.59	14.86	41.62	45.95	12.43
V7	15.41	76.76	7.84	48.38	30.81	20.81
OSV7	23.78	72.16	4.05	50.27	35.14	14.59
Ideal	33.78	65.95	0.27	52.97	47.03	0.00

in Fig. 79. From the distribution and the plots, it can be seen that the predictive path switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXXII. PESQ-MOS distributions calculated over every 0.5 MOS point for #4 trace-file path pair $A_{new}B_{new}$.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A_{new}	7.30	1.35	2.16	3.24	4.32	11.35	70.27
Path B_{new}	14.05	4.59	7.57	6.22	8.65	7.57	51.35
SP_{CLR}	0.81	1.35	1.35	4.05	8.65	8.92	74.86
V7	0.54	0.81	2.43	3.51	8.38	14.59	69.73
OSV7	0.27	0.27	0.81	1.62	4.86	14.86	77.30
Ideal	0.00	0.27	0.54	0.27	2.16	2.43	94.32

The cumulative PESQ-MOS distribution is plotted in Fig. 80. Below MOS level of 3, the plots of predictive path switching controls are lower (better) than the no switching methods. At MOS level of 3.5, plots of both SP_{CLR} control and V7 control match that of Path A_{new} method, which indicates that predictive path switching controls improve the bad quality segments (below MOS level of 2) to the fair quality range (around MOS level of 3), but have difficulty in improving them into the good quality level (above MOS level of 3.5). The plot of OSV7 control is lower (better)

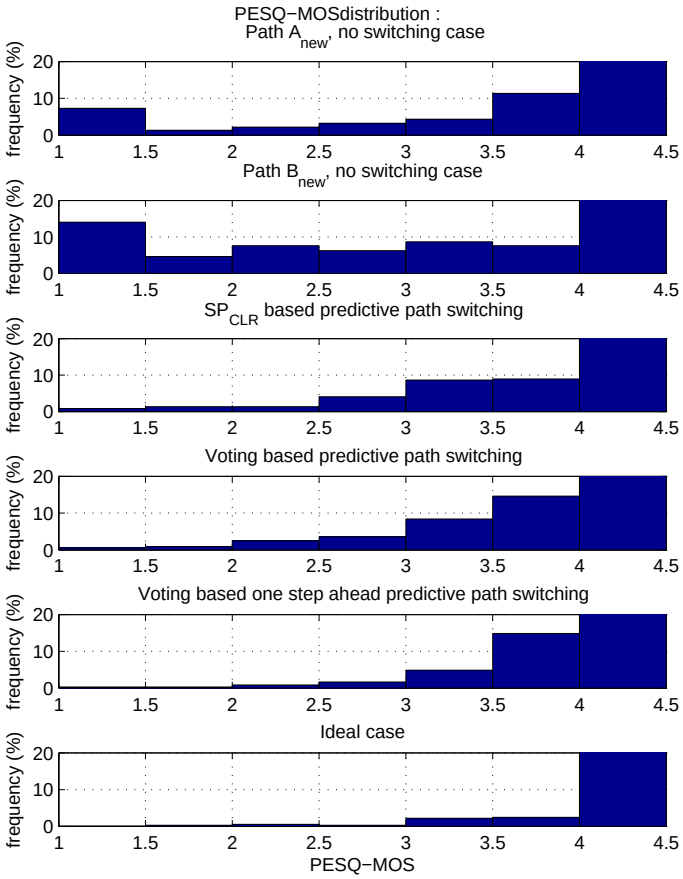

Fig. 79. Zoom-in PESQ-MOS distribution of #4 trace-file path pair $\mathrm{A_{new}B_{new}}$.

than SP_{CLR} control and V7 control. The plot of Ideal control is the lowest (best).

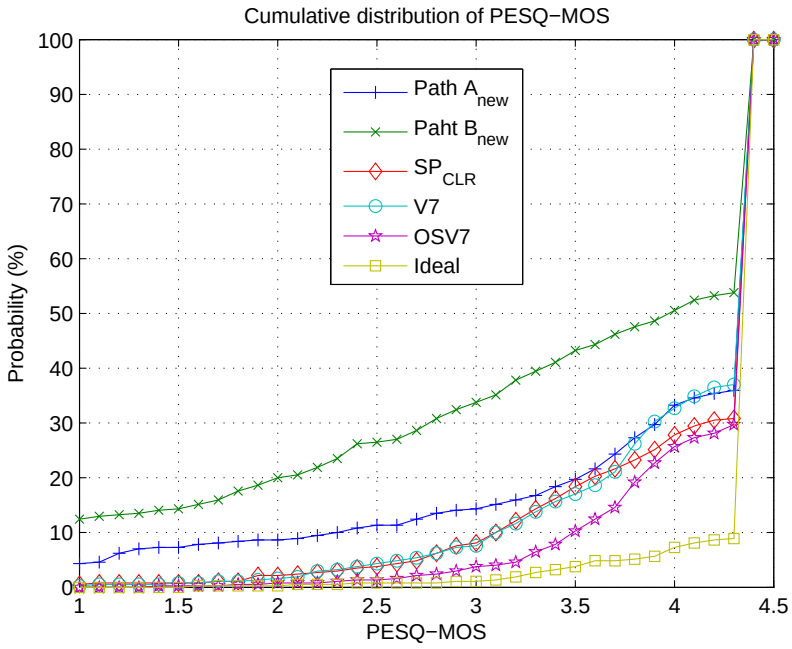

Fig. 80. Cumulative PESQ-MOS distribution of #4 trace-file path pair $A_{new}B_{new}$.

Table LXXIII presents the resulting average CLR and PESQ-MOS of the different controls for the 8th trace-file of path pair $A_{new}B_{new}$. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A_{new} method (increments of 0.19 and 0.16, respectively) and Path B_{new} method (increments of 1.02 and 0.98, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.24 and 1.07 over Path A_{new} method and Path B_{new} method, respectively) and Ideal control (increments of 0.38 and 1.21 over Path A_{new} method and Path B_{new} method, respectively).

Table LXXIII. Average CLR and PESQ-MOS of #8 trace-file path pair $A_{new}B_{new}$.

	Path A_{new}	Path B_{new}	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	6.23	23.47	2.58	2.74	2.03	1.39
SD CLR (%)	3.73	27.05	1.78	1.46	1.53	1.62
Avg. PESQ-MOS	3.91	3.08	4.10	4.06	4.15	4.29
SD PESQ-MOS	1.02	1.49	0.76	0.76	0.68	0.60

Table LXXIV shows the voice quality distribution of each control for the 8th trace-file of path pair $A_{new}B_{new}$. It shows that SP_{CLR} control is better than both Path A_{new} method (10.27% bad quality segments vs 15.41% bad quality segments), and Path B_{new} method (10.27% bad quality segments vs 43.24% bad quality segments), V7 control is better than both Path A_{new} method (10.27% bad segments vs 15.41% bad segmetns) and Path B_{new} method (10.27% bad segments vs 43.24% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (7.30% bad segments). The Ideal control gives the best result (5.68% bad segments).

Table LXXIV. Voice quality distribution of #8 trace-file path pair $A_{new}B_{new}$.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A_{new}	15.41	7.57	77.03
B_{new}	43.24	5.14	51.62
SP_{CLR}	10.27	7.30	82.43
V7	10.27	8.38	81.35
OSV7	7.30	4.86	87.84
Ideal	5.68	3.24	91.08

Table LXXV gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 8^{th} trace-file of path pair $A_{new}B_{new}$. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A_{new} method and Path B_{new} method are positive, they improve the voice quality compared to Path A_{new} method and Path B_{new} method. The OSV7 control results are better than SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXXVI is the comparison of the predictive path switching control and the no switching methods for the 8^{th} trace-file of path pair $A_{new}B_{new}$. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A_{new} method and Path B_{new} method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXXVII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 8^{th} trace-file of path pair $A_{new}B_{new}$. A zoom-in version is plotted in Fig. 81. From the distribution and the plots, it can be seen that the predictive path

Table LXXV. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #8 trace-file path pair $A_{new}B_{new}$.

	A	B	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	15.41	43.24	10.27	10.27	7.30	5.68
Improvement over path A (%)			33.33	33.33	52.63	63.16
Improvement over path B (%)			76.25	76.25	83.13	86.88
Improvement over SP_{CLR} (%)				0.00	28.95	44.74
PESQ-MOS < 3.5 (%)	22.97	48.38	17.57	18.65	12.16	8.92
Improvement over path A (%)			23.53	18.82	47.06	61.18
Improvement over path B (%)			63.69	61.45	74.86	81.56
Improvement over SP_{CLR} (%)				-6.15	30.77	49.23

Table LXXVI. Comparison of predictive path switching control and no switching methods for #8 trace-file path pair $A_{new}B_{new}$.

	Comparing to path A_{new} (%)			Comparing to path B_{new} (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	18.11	71.08	10.81	48.65	46.49	4.86
V7	14.86	77.84	7.30	51.62	34.32	14.05
OSV7	20.00	75.68	4.32	52.16	35.95	11.89
Ideal	28.65	70.81	0.54	53.51	46.22	0.27

switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXXVII. PESQ-MOS distributions calculated over every 0.5 MOS point for #8 trace-file path pair $A_{new}B_{new}$.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A_{new}	7.57	1.89	2.70	3.24	7.57	8.65	68.38
Path B_{new}	27.84	6.76	5.41	3.24	5.14	5.68	45.95
SP_{CLR}	1.08	1.62	4.05	3.51	7.30	7.84	74.59
V7	0.81	1.89	3.78	3.78	8.38	10.81	70.54
OSV7	1.08	1.35	2.43	2.43	4.86	11.89	75.95
Ideal	0.54	1.08	2.16	1.89	3.24	2.97	88.11

The cumulative PESQ-MOS distribution is plotted in Fig. 82. Here, the plots of predictive path switching controls are lower (better) than the no switching methods. There are some distances at both MOS level of 3, and MOS level of 3.5, which indicates the predictive path switching controls give improvement to both the bad quality segments (below MOS level of 2) and the fair quality segments (around MOS level of 3). The plot of OSV7 control is lower (better) than V7 control. The SP_{CLR} control is lower (better) than both V7 control and OSV7 above MOS level of 4 here. The plot of Ideal control is the lowest (best).

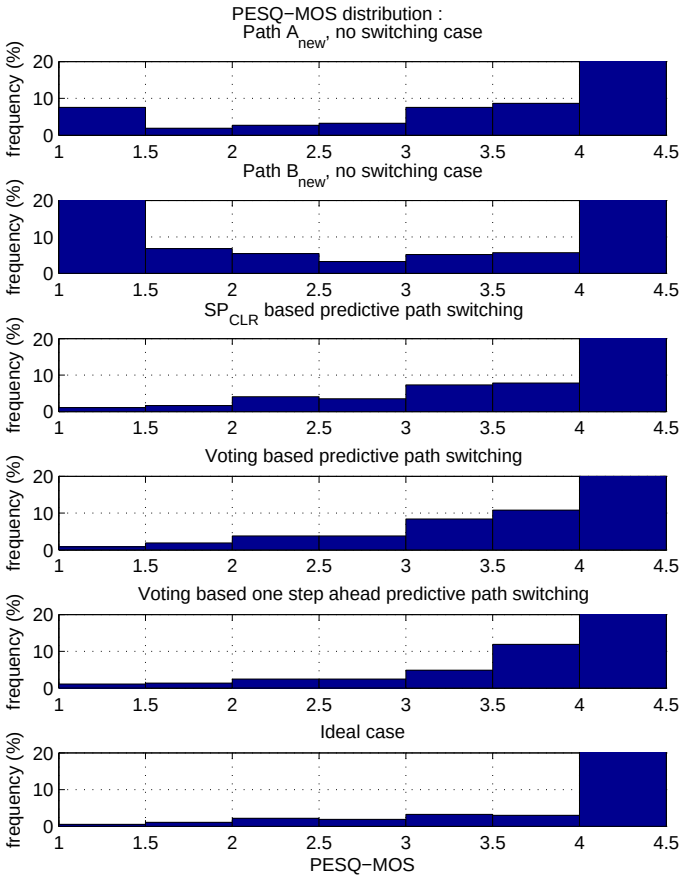

Fig. 81. Zoom-in PESQ-MOS distribution of #8 trace-file path pair $\mathrm{A_{new}B_{new}}$.

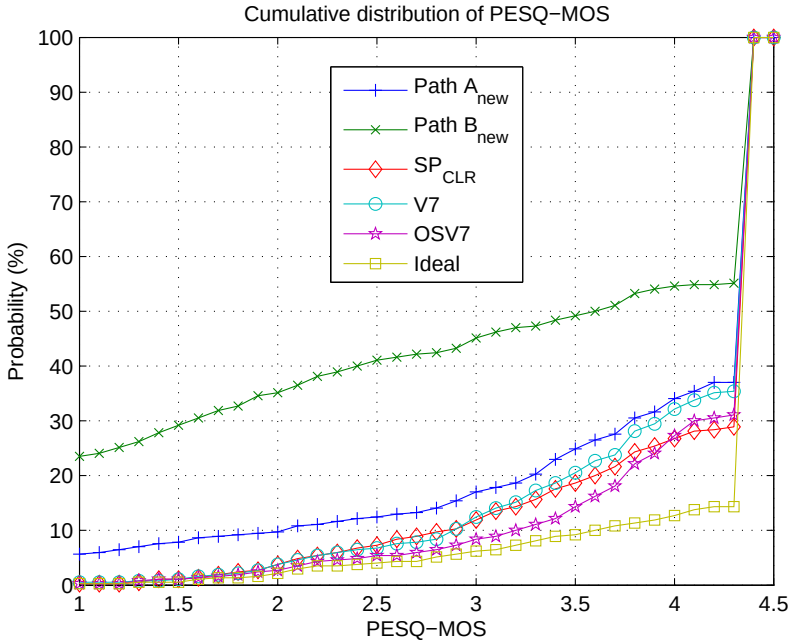

Fig. 82. Cumulative PESQ-MOS distribution of #8 trace-file path pair $\mathrm{A_{new}B_{new}}$.

Table LXXVIII presents the resulting average CLR and PESQ-MOS of the different controls for the 4th trace-file of path pair $A_{new}C_{new}$. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A_{new} method (increments of 0.28 and 0.17, respectively) and Path C_{new} method (increments of 0.27 and 0.16, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.42 and 0.31 over Path A_{new} method and Path C_{new} method, respectively) and Ideal control (increments of 0.52 and 0.41 over Path A_{new} method and Path C_{new} method, respectively).

Table LXXVIII. Average CLR and PESQ-MOS of #4 trace-file path pair $A_{new}C_{new}$.

	Path A_{new}	Path C_{new}	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	5.61	4.37	1.58	1.42	0.64	0.18
SD CLR (%)	2.13	1.25	0.29	0.45	0.33	0.21
Avg. PESQ-MOS	3.94	4.05	4.22	4.21	4.36	4.46
SD PESQ-MOS	1.02	0.90	0.57	0.54	0.35	0.21

Table LXXIX shows the voice quality distribution of each control for the 4th trace-file of path pair $A_{new}C_{new}$. It shows that SP_{CLR} control is better than both Path A_{new} method (7.03% bad quality segments vs 14.05% bad quality segments), and Path C_{new} method (7.03% bad quality segments vs 13.78% bad quality segments), V7 control is better than both Path A_{new} method (5.68% bad segments vs 14.05% bad segmetns) and Path C_{new} method (5.68% bad segments vs 13.78% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (1.35% bad segments). The Ideal control gives the best result (0.81% bad segments).

Table LXXIX. Voice quality distribution of #4 trace-file path pair $A_{new}C_{new}$.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A_{new}	14.05	4.32	81.62
C_{new}	13.78	4.86	81.35
SP_{CLR}	7.03	5.95	87.03
V7	5.68	5.95	88.38
OSV7	1.35	1.62	97.03
Ideal	0.81	0.54	98.65

Table LXXX gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 4^{th} trace-file of path pair $A_{new}C_{new}$. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A_{new} method and Path C_{new} method are positive, they improve the voice quality compared to Path A_{new} method and Path C_{new} method. The OSV7 control results are better than SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXXXI is the comparison of the predictive path switching control and the no switching methods for the 4^{th} trace-file of path pair $A_{new}C_{new}$. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A_{new} method and Path C_{new} method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXXXII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 4^{th} trace-file of path pair $A_{new}C_{new}$. A zoom-in version is plotted in Fig. 83. From the distribution and the plots, it can be seen that the predictive path

Table LXXX. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #4 trace-file path pair $A_{new}C_{new}$.

	A_{new}	C_{new}	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	14.05	13.78	7.03	5.68	1.35	0.81
Improvement over path A_{new} (%)			50.00	59.62	90.38	94.23
Improvement over path C_{new} (%)			49.02	58.82	90.20	94.12
Improvement over SP_{CLR} (%)				19.23	80.77	88.46
PESQ-MOS < 3.5 (%)	18.38	18.65	12.97	11.62	2.97	1.35
Improvement over path A_{new} (%)			29.41	36.76	83.82	92.65
Improvement over path C_{new} (%)			30.43	37.68	84.06	92.75
Improvement over SP_{CLR} (%)				10.42	77.08	89.58

Table LXXXI. Comparison of predictive path switching control and no switching methods for #4 trace-file path pair $A_{new}C_{new}$.

	Comparing to path A_{new} (%)			Comparing to path C_{new} (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	27.03	62.16	10.81	20.00	69.19	10.81
V7	26.76	59.73	13.51	21.62	66.76	11.62
OSV7	30.81	60.81	8.38	24.05	70.54	5.41
Ideal	35.14	64.59	0.27	27.30	72.16	0.54

switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXXXII. PESQ-MOS distributions calculated over every 0.5 MOS point for #4 trace-file path pair $A_{new}C_{new}$.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A_{new}	7.30	1.35	2.16	3.24	4.32	11.35	70.27
Path C_{new}	5.14	1.89	2.43	4.32	4.86	7.03	74.32
SP_{CLR}	0.00	0.81	1.08	5.14	5.95	8.11	78.92
V7	0.00	0.81	1.35	3.51	5.95	12.97	75.41
OSV7	0.00	0.00	0.81	0.54	1.62	10.54	86.49
Ideal	0.00	0.00	0.27	0.54	0.54	1.08	97.57

The cumulative PESQ-MOS distribution is plotted in Fig. 84. Below MOS level of 3.5, the plots of predictive path switching controls are lower (better) than the no switching methods. Above MOS level of 3.7 plot of V7 control matches or is higher (worse) that of Path C_{new} method, plot of both SP_{CLR} control is also close to that of Path C_{new} method, which indicates that predictive path switching controls improve the bad quality segments (below MOS level of 2) to the fair quality range (around MOS level of 3), but have difficulty in improving them into the good quality level (above MOS level of 3.5). The plot of OSV7 control is lower (better) than SP_{CLR}

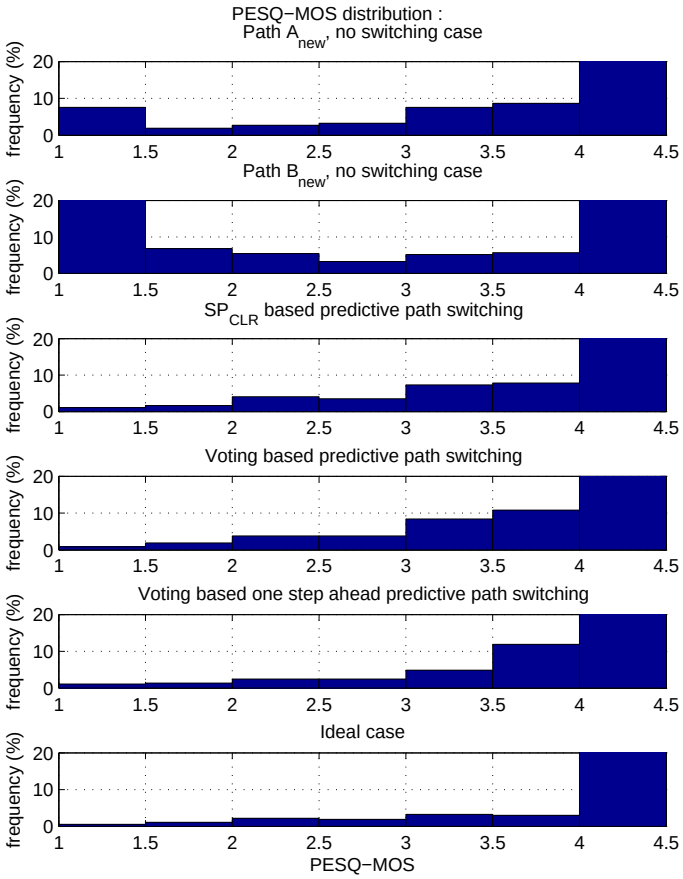

Fig. 83. Zoom-in PESQ-MOS distribution of #4 trace-file path pair $\mathrm{A_{new}C_{new}}$.

control and V7 control. The plot of Ideal control is the lowest (best).

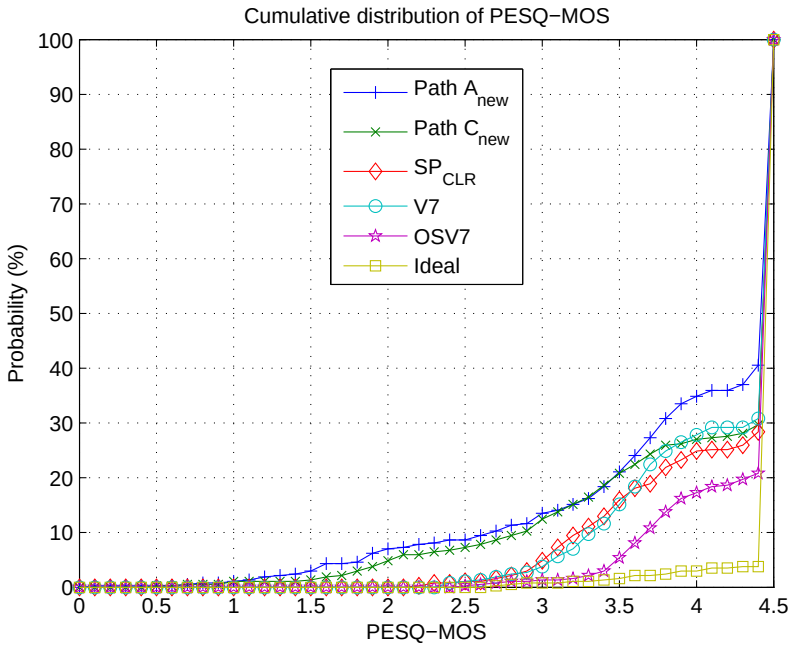

Fig. 84. Cumulative PESQ-MOS distribution of #4 trace-file path pair $A_{new}C_{new}$.

Table LXXXIII presents the resulting average CLR and PESQ-MOS of the different controls for the 8th trace-file of path pair $A_{new}C_{new}$. It is observed that SP_{CLR} control and V7 control increase (improve) the resulting average PESQ-MOS compared to Path A_{new} method (increments of 0.24 and 0.26, respectively) and Path C_{new} method (increments of 0.26 and 0.28, respectively). But the increments (improvements) are not as much as those of OSV7 control (increments of 0.42 and 0.44 over Path A_{new} method and Path C_{new} method, respectively) and Ideal control (increments of 0.52 and 0.54 over Path A_{new} method and Path C_{new} method, respectively).

Table LXXXIII. Average CLR and PESQ-MOS of #8 trace-file path pair $A_{new}C_{new}$.

	Path A_{new}	Path C_{new}	SP_{CLR}	V7	OSV7	Ideal
Avg. CLR (%)	6.23	6.88	2.04	1.80	0.86	0.43
SD CLR (%)	3.73	3.42	0.86	0.91	0.77	0.63
Avg. PESQ-MOS	3.91	3.89	4.15	4.17	4.33	4.43
SD PESQ-MOS	1.02	1.10	0.65	0.59	0.45	0.35

Table LXXXIV shows the voice quality distribution of each control for the 8th trace-file of path pair $A_{new}C_{new}$. It shows that SP_{CLR} control is better than both Path A_{new} method (8.65% bad quality segments vs 15.41% bad quality segments), and Path C_{new} method (8.65% bad quality segments vs 17.30% bad quality segments), V7 control is better than both Path A_{new} method (6.22% bad segments vs 15.41% bad segmetns) and Path C_{new} method (6.22% bad segments vs 17.30% bad segments). Compared to no switching methods, predictive path switching controls reduce the percentage of bad segments. If there is no feedback delay, then there is more improvement in voice quality as shown by the OSV7 control results (2.16% bad segments). The Ideal control gives the best result (1.62% bad segments).

Table LXXXIV. Voice quality distribution of #8 trace-file path pair $A_{new}C_{new}$.

Quality	Bad(%)	Fair(%)	Good(%)
PESQ-MOS	< 3	$3 \sim 3.5$	> 3.5
A_{new}	15.41	7.57	77.03
C_{new}	17.30	6.22	76.49
SP_{CLR}	8.65	8.38	82.97
V7	6.22	7.57	86.22
OSV7	2.16	1.89	95.95
Ideal	1.62	0.54	97.84

Table LXXXV gives the percentage of segment that has voice quality below 3 of each control, and the percentage of segment that has voice quality below 3.5 of each control for the 8^{th} trace-file of path pair $A_{new}C_{new}$. It shows that the percentage improvements of SP_{CLR} control and V7 control over Path A_{new} method and Path C_{new} method are positive, they improve the voice quality compared to Path A_{new} method and Path C_{new} method. The OSV7 control results are better than SP_{CLR} control and V7 control. The ideal case predictive path switching control gives the best results.

Table LXXXVI is the comparison of the predictive path switching control and the no switching methods for the 8^{th} trace-file of path pair $A_{new}C_{new}$. It can be seen that ideal case predictive path switching control will either improve the result or keep it the same, there will hardly be any case of being worse than no switching control. Compared to Path A_{new} method and Path C_{new} method, there are more improvements from SP_{CLR} control and V7 control than degradations.

Table LXXXVII presents the PESQ-MOS distributions calculated over every 0.5 MOS point for the 8^{th} trace-file of path pair $A_{new}C_{new}$. A zoom-in version is plotted in Fig. 85. From the distribution and the plots, it can be seen that the predictive path

Table LXXXV. Percentage of PESQ-MOS below 3 and percentage of PESQ-MOS below 3.5 of different cases for #8 trace-file path pair $A_{new}C_{new}$.

	A_{new}	C_{new}	SP_{CLR}	V7	OSV7	Ideal
PESQ-MOS < 3 (%)	15.41	17.30	8.65	6.22	2.16	1.62
Improvement over path A_{new} (%)			43.86	59.65	85.96	89.47
Improvement over path C_{new} (%)			50.00	64.06	87.50	90.63
Improvement over SP_{CLR} (%)				28.13	75.00	81.25
PESQ-MOS < 3.5 (%)	22.97	23.51	17.03	13.78	4.05	2.16
Improvement over path A_{new} (%)			25.88	40.00	82.35	90.59
Improvement over path C_{new} (%)			27.59	41.38	82.76	90.80
Improvement over SP_{CLR} (%)				19.05	76.19	87.30

Table LXXXVI. Comparison of predictive path switching control and no switching methods for #8 trace-file path pair $A_{new}C_{new}$.

	Comparing to path A_{new} (%)			Comparing to path C_{new} (%)		
method	Better	Same	Worse	Better	Same	Worse
SP_{CLR}	24.32	63.51	12.16	21.08	65.41	13.51
V7	24.86	61.08	14.05	22.70	62.43	14.86
OSV7	31.35	62.43	6.22	27.30	65.41	7.30
Ideal	35.14	64.59	0.27	30.00	69.73	0.27

switching controls reduce the extremely bad quality segments, where PESQ-MOSs are less than 2, and improved them to the fair quality range, where PESQ-MOSs are around 3. But they cannot further improve them into the good quality range, which requires the PESQ-MOS to be more than 3.5. If there is no feedback delay then some segments can be improved into the good quality range. And in the ideal case predictive path switching control many segments can be improved into the good quality range.

Table LXXXVII. PESQ-MOS distributions calculated over every 0.5 MOS point for #8 trace-file path pair $A_{new}C_{new}$.

Method	Percentage in PESQ-MOS range						
	$1 \sim 1.5$	$1.5 \sim 2$	$2 \sim 2.5$	$2.5 \sim 3$	$3 \sim 3.5$	$3.5 \sim 4$	$4 \sim 4.5$
Path A_{new}	7.57	1.89	2.70	3.24	7.57	8.65	68.38
Path C_{new}	8.38	2.97	4.32	1.62	6.22	3.78	72.70
SP_{CLR}	0.54	0.54	1.62	5.95	8.38	7.03	75.95
V7	0.81	0.54	0.81	4.05	7.57	11.89	74.32
OSV7	0.54	0.54	0.27	0.81	1.89	11.35	84.59
Ideal	0.54	0.54	0.00	0.54	0.54	2.70	95.14

The cumulative PESQ-MOS distribution is plotted in Fig. 86. Below MOS level of 3.5, the plots of predictive path switching controls are lower (better) than the no switching methods. Above MOS level of 3.7 plot of V7 control matches or is higher (worse) that of Path C_{new} method, plot of both SP_{CLR} control is also close to that of Path C_{new} method, which indicates that predictive path switching controls improve the bad quality segments (below MOS level of 2) to the fair quality range (around MOS level of 3), but have difficulty in improving them into the good quality level (above MOS level of 3.5). The plot of OSV7 control is lower (better) than SP_{CLR}

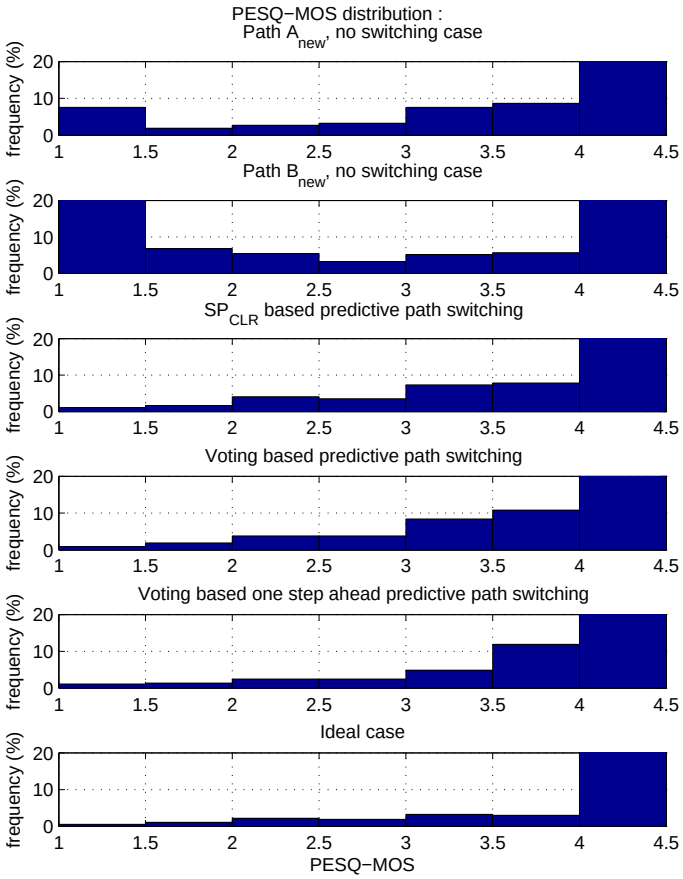

Fig. 85. Zoom-in PESQ-MOS distribution of #8 trace-file path pair $\mathrm{A_{new}C_{new}}$.

control and V7 control. The plot of Ideal control is the lowest (best).

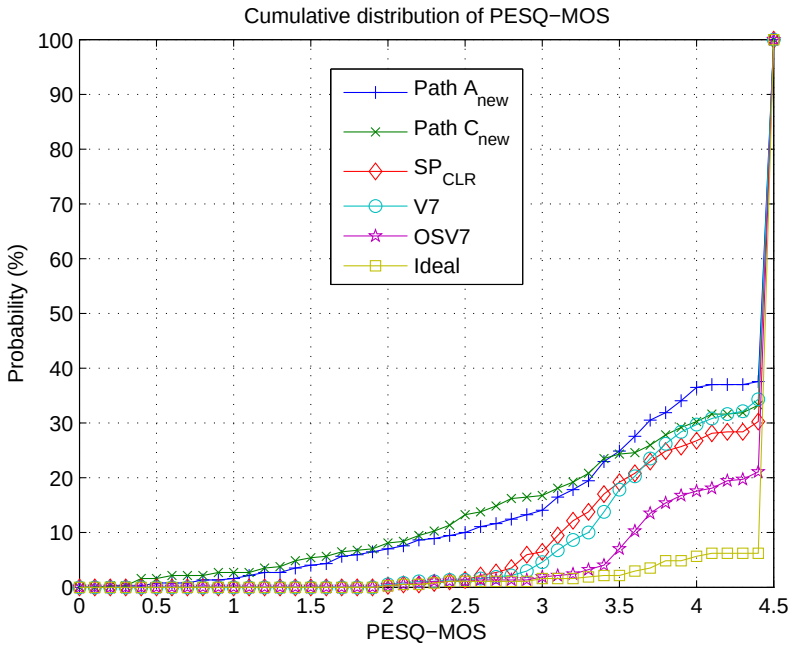

Fig. 86. Cumulative PESQ-MOS distribution of #8 trace-file path pair $A_{new}C_{new}$.

VITA

Dan Ye received his Bachelor of Science degree in mechanical engineering and automation and his Master of Science degree in precision instruments from Tsinghua University, Beijing, P.R. China, in 2000 and 2002, respectively. He received his Doctor of Philosophy degree in mechanical engineering from Texas A&M University in 2006. His research interests include biomedical image processing and identification, adaptive modeling and flow control of distributed real-time applications, and dynamic system estimation and control.

His publications include

1. A. G. Parlos and D. Ye, "Transport model for flow processes characterized by time-varying time delays," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 9-12 Dec. 2003, vol. 3, pp. 2618 - 2619.

2. G. Qi, D. Ye, Q. Hong, and D. Xu, "Blood cells image decomposition based on circular assumption and polygonal approximation," *Computer Engineering and Application*, 2003, vol. 39, no. 16, pp. 75-77, 85.

3. Q. Hong, G. Qi, and D. Ye, "MDB/ICP protocol based master controller realization of an automatic vending system," *Application of Electronic Technique*, 2002, vol. 28, no. 9, pp 28-30.

4. D. Ye, G. Qi, Q. Hong, and X. Li, "Single board computer based automatic temperature control system," in *Journal of Transducer Technology*, 2002, vol. 21, no. 3, pp 27-30.

Mr. Ye may be reached at Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843. His email address is dan_ye@tamu.edu.