

PREDICTOR DEVELOPMENT FOR CONTROLLING  
REAL-TIME APPLICATIONS OVER THE INTERNET

A Thesis

by

MALLIK KOMMARAJU

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

MASTER OF SCIENCE

December 2005

Major Subject: Mechanical Engineering

PREDICTOR DEVELOPMENT FOR CONTROLLING  
REAL-TIME APPLICATIONS OVER THE INTERNET

A Thesis

by

MALLIK KOMMARAJU

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

MASTER OF SCIENCE

Approved by:

Chair of Committee, Alexander Parlos  
Committee Members, Won-jong Kim  
Dmitri Loguinov

Head of Department, Dennis O'Neal

December 2005

Major Subject: Mechanical Engineering

## ABSTRACT

Predictor Development for Controlling Real-Time Applications over the Internet.

(December 2005)

Mallik Kommaraju, B.Tech., Indian Institute of Technology, Madras

Chair of Advisory Committee: Dr. Alexander Parlos

Over the past decade there has been a growing demand for interactive multimedia applications deployed over public IP networks. To achieve acceptable Quality of Service (QoS) without significantly modifying the existing infrastructure, the end-to-end applications need to optimize their behavior and adapt according to network characteristics. Most existing application optimization techniques are based on reactive strategies, i.e. reacting to occurrences of congestion. We propose the use of predictive control to address the problem in an anticipatory manner. This research deals with developing models to predict end-to-end single flow characteristics of Wide Area Networks (WANs).

A novel signal, in the form of single flow packet accumulation, is proposed for feedback purposes. This thesis presents a variety of effective predictors for the above signal using Auto-Regressive (AR) models, Radial Basis Functions (RBF) and Sparse Basis Functions (SBF). The study consists of three sections. We first develop time-series models to predict the accumulation signal. Since encoder bit-rate is the most logical and generic control input, a statistical analysis is conducted to analyze the effect of input bit-rate on end-to-end delay and the accumulation signal. Finally, models are developed using this bit-rate as an input to predict the resulting accumulation signal. The predictors are evaluated based on Noise-to-Signal Ratio (NSR) along with their accuracy with increasing accumulation levels. In time-series models,

RBF gave the best NSR closely followed by AR models. Analysis based on accuracy with increasing accumulation levels showed AR to be better in some cases. The study on effect of bit-rate revealed that bit-rate may not be a good control input on all paths. Models such as Auto-Regressive with Exogenous input (ARX) and RBF were used to develop models to predict the accumulation signal using bit-rate as a modeling input. ARX and RBF models were found to give comparable accuracy, with RBF being slightly better.

To My Parents

## ACKNOWLEDGMENTS

I would like to convey my sincere thanks and gratitude to my committee chair and advisor, Dr. Alexander G. Parlos, for his patience, continuous guidance, technical support and advice through the course of my research work. I also thank Dr. Wonjong Kim and Dr. Dmitri Loguinov for their time, interest and support in my research.

Finally, I would like to acknowledge and thank all the students at NIML, especially, Aninda, Dan Ye and Parasuram, for several insightful discussions on related and unrelated topics.

## TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION . . . . .	1
	A. Research Objectives . . . . .	3
	B. Literature Review . . . . .	4
	1. End-to-End Analysis . . . . .	4
	2. Network Simulators . . . . .	4
	3. Predictability Analysis . . . . .	5
	4. System Identification as Applied to the Internet . . . . .	5
	5. Neural Networks and Fuzzy Logic . . . . .	6
	6. Accumulation as a Feedback Signal . . . . .	7
	C. Proposed Approach . . . . .	7
	D. Contributions of the Current Research . . . . .	8
	E. Organization of the Thesis . . . . .	9
II	MODELING AND PREDICTION TECHNIQUES . . . . .	10
	A. Introduction . . . . .	10
	B. Modeling Categories . . . . .	10
	C. System Identification Procedure . . . . .	11
	D. Linear Model Structures . . . . .	14
	1. Auto-Regressive and Auto-Regressive with Exogenous Input Models . . . . .	17
	2. Auto-Regressive Moving Average with Exogenous Input Model . . . . .	18
	E. Non-Linear Model Structures . . . . .	19
	1. Radial Basis Functions (RBF) . . . . .	19
	2. Sparse Basis Functions . . . . .	21
	F. Chapter Summary . . . . .	22
III	TIME-SERIES PREDICTION OF ACCUMULATION SIGNAL	23
	A. Introduction . . . . .	23
	1. Experimental Set-up . . . . .	23
	2. Ns-2 Simulation Setup . . . . .	23
	3. Real-world Network Experiments . . . . .	24
	a. Experiment Description . . . . .	25

CHAPTER	Page
B. Accumulation Signal . . . . .	26
C. Error Calculation Technique . . . . .	27
D. Experiment and Model Specifications . . . . .	28
E. Autoregressive (AR) Models . . . . .	29
1. Parameter Selection . . . . .	29
2. AR and SP Prediction Results . . . . .	32
F. Radial Basis Function (RBF) Models . . . . .	40
1. Parameter Selection . . . . .	40
2. Radial Basis Function Prediction Results . . . . .	42
G. Sparse Basis Functions . . . . .	46
1. Parameter Selection . . . . .	46
2. Sparse Basis Function Prediction Results . . . . .	52
H. Chapter Summary . . . . .	52
IV EFFECT OF BIT-RATE ON ACCUMULATION SIGNAL . . . . .	58
A. Introduction . . . . .	58
B. Statistical Analysis . . . . .	60
1. Experiments on Path-1 . . . . .	63
2. Experiments on Path-3 . . . . .	65
C. Prediction of Accumulation Signal . . . . .	68
1. Parameter Selection . . . . .	69
2. Results for the Prediction of Accumulation in Bytes . . . . .	72
a. Experiments on Path-1 . . . . .	72
b. Experiments on Path-3 . . . . .	77
3. Results for Prediction of Accumulation in Packets . . . . .	88
a. Experiment 1 . . . . .	88
b. Experiments on Path-3 . . . . .	98
D. Chapter Summary . . . . .	99
V SUMMARY AND CONCLUSIONS . . . . .	108
A. Summary . . . . .	108
B. Conclusions and Recommendations . . . . .	114
REFERENCES . . . . .	116
VITA . . . . .	119



## LIST OF TABLES

TABLE		Page
I	Description of Constant Bit-Rate Experiments . . . . .	30
II	Model Parameters for Time-Series Prediction . . . . .	31
III	Description of Variable Bit-Rate Experiments . . . . .	61
IV	Model Parameters (Accumulation in Bytes) . . . . .	73
V	Model Parameters (Accumulation in Packets) . . . . .	74

## LIST OF FIGURES

FIGURE		Page
1	System identification procedure. . . . .	12
2	General linear model. . . . .	15
3	Linear time-series models. . . . .	15
4	Linear input-output models. . . . .	16
5	Schematic diagram of RBF networks. . . . .	20
6	General experimental set-up. . . . .	24
7	ns-2 simulation set-up. . . . .	25
8	Auto-correlation functions for ns-2 accumulation data. . . . .	33
9	Auto-correlation functions for path-1 accumulation signal. . . . .	33
10	Auto-correlation functions for path-2 accumulation signal. . . . .	34
11	AIC criteria for path-1 accumulation signal. . . . .	34
12	AIC criteria for path-2 accumulation signal. . . . .	35
13	Comparison of AR, RBF and SBF predictors for accumulation signal on path-1. . . . .	36
14	Comparison of AR, RBF and SBF predictors for accumulation signal on path-1. . . . .	37
15	Comparison of AR, RBF1, RBF2 and SBF predictors for accu- mulation signal on path-2. . . . .	38
16	Comparison of AR and SP for ns-2 accumulation signal. . . . .	39
17	Comparison of AR and SP for path-1 accumulation signal. . . . .	39

FIGURE	Page
18	Comparison of AR and SP for path-2 accumulation signal. . . . . 40
19	Parameter selection for RBF network, 0.5 sec prediction NSR for different number of neurons and for different radii for the Gaussian function. . . . . 41
20	Comparison of RBF and AR predictors for ns-2 simulation. . . . . 44
21	Comparison of RBF and AR predictors on path-1. . . . . 44
22	Comparison of RBF and AR (order 350) predictors on path-2. . . . . 45
23	Comparison of RBF1 and AR (order 40) predictors on path-2. . . . . 45
24	Comparison of RBF2 and AR (order 40) predictors on path-2. . . . . 46
25	0.5 second prediction using RBF and AR predictors for ns-2 simulation scenario. . . . . 47
26	0.5 second prediction using RBF and AR predictors on path-1. . . . . 48
27	0.5 second prediction using RBF and AR predictors on path-2. . . . . 48
28	0.5 second prediction using RBF2 and AR predictors on path-2. . . . . 49
29	0.5 second prediction using RBF2 and AR predictors on path-2, another segment of the signal. . . . . 50
30	Effect of the number of initialization points and the forgetting factor on the predictor <i>NSR</i> using sparse basis functions. . . . . 51
31	SBF vs SP on ns-2 simulated data. . . . . 53
32	SBF vs SP on path-1. . . . . 53
33	SBF vs SP on path-2. . . . . 54
34	SBF vs AR predictor on path-2. . . . . 54
35	0.5 second prediction using SBF and SP Predictors for ns-2 simulation scenario. . . . . 55

FIGURE	Page
36	0.5 second prediction using SBF and SP Predictors on path-1. . . . . 56
37	0.5 second prediction using SBF and SP Predictors on path-2. . . . . 56
38	Adjusted bit-rate vs time for experiments to explore the impact of bit-rate on packet loss and delayed packets. . . . . 59
39	One-way delay traces for path-1. . . . . 60
40	One-way delay traces for path-3. . . . . 62
41	CDF for packet loss in the experiment on path-1. . . . . 64
42	CDF for total loss in the experiment on path-1. . . . . 64
43	Scatter plot for total loss when using two bit-rates in the experi- ment on path-1. . . . . 65
44	Time-ensemble average of the one-way delay on path-1. . . . . 66
45	Time-ensemble average of the total loss on path-1. . . . . 66
46	CDF for packet loss in the experiment on path-3. . . . . 67
47	CDF for total loss in the experiment on path-3. . . . . 68
48	Scatter plot for total loss in the experiment on path-3. . . . . 69
49	Time-ensemble average of the one-way delay on path-3. . . . . 70
50	Time-ensemble average of the total loss on path-3. . . . . 71
51	Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-1 with accumulation in bytes. . . . . 72
52	Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-3 with accumulation in bytes. . . . . 73
53	Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-1 with accumulation in packets. . . . . 74

FIGURE	Page
54	Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-3 with accumulation in packets). . . . . 75
55	Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for path-1. . . . . 77
56	Prediction of byte-accumulation with AR and SP on path-1. . . . . 78
57	Prediction of byte-accumulation with ARMA and AR on path-1. . . 78
58	Prediction of byte-accumulation with ARX and AR on path-1. . . . . 79
59	Prediction of byte-accumulation with ARMMAX and AR on path-1. 79
60	Prediction of byte-accumulation with RBF and AR on path-1. . . . . 80
61	0.5 sec prediction of byte-accumulation with AR and ARX predictors on path-1. . . . . 81
62	0.5 sec prediction of byte-accumulation with ARMA and ARMAX predictors on path-1. . . . . 82
63	0.5 sec prediction of byte-accumulation with AR and RBF predictors on path-1. . . . . 83
64	Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for accumulation in bytes on path-3. . . 84
65	Prediction of byte-accumulation with AR and SP on path-3. . . . . 85
66	Prediction of byte-accumulation with ARMA and AR on path-3 . . . 85
67	Prediction of byte-accumulation with ARX and AR on path-3. . . . . 86
68	Prediction of byte-accumulation with ARMMAX and AR on path-3. 87
69	Prediction of byte-accumulation with RBF and AR on path-3. . . . . 87
70	0.5 sec prediction of byte-accumulation with AR and ARX predictors on path-3. . . . . 88

FIGURE	Page
71	0.5 sec prediction of byte-accumulation with ARMA and ARMAX predictors on path-3. . . . . 89
72	0.5 sec prediction of byte-accumulation with AR and RBF predictors on path-3. . . . . 90
73	Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon on path-1. . . . . 92
74	Prediction of packet-accumulation with AR and SP on path-1. . . . . 93
75	Prediction of packet-accumulation with ARMA and AR on path-1. . . . . 93
76	Prediction of packet-accumulation with ARX and AR on path-1. . . . . 94
77	Prediction of packet-accumulation with ARMMAX and AR on path-1. . . . . 94
78	Prediction of packet-accumulation with RBF and AR on path-1. . . . . 95
79	0.5 sec prediction of packet-accumulation with AR and ARX on path-1. . . . . 95
80	0.5 sec prediction of packet-accumulation with ARMA and ARMAX predictors on path-1. . . . . 96
81	0.5 sec prediction of packet-accumulation with AR and RBF predictors on path-1. . . . . 97
82	Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for accumulation in packets on path-3. . . . . 99
83	Prediction of packet-accumulation with AR and SP on path-3. . . . . 100
84	Prediction of packet-accumulation with ARMA and AR on path-3. . . . . 100
85	Prediction of packet-accumulation with ARX and AR on path-3. . . . . 101
86	Prediction of packet-accumulation with ARMMAX and AR on path-3. . . . . 101
87	Prediction of packet-accumulation with RBF and AR on path-3. . . . . 102

FIGURE		Page
88	0.5 sec prediction of packet-accumulation with AR and ARX models on path-3. . . . .	103
89	0.5 sec prediction of packet-accumulation with ARMA and ARMAX models on path-3. . . . .	104
90	0.5 sec prediction of packet-accumulation with RBF and AR on path-3. . . . .	105

## CHAPTER I

### INTRODUCTION

Increasing demand for multi-media and real-time applications across the Internet has exposed some of the deficiencies in the design of Wide Area Networks (WAN). The current network architecture was implemented as a best effort network and is suitable for non-real-time applications. Modifying the existing infrastructure to accommodate real-time applications is not a feasible solution due to economic considerations. Therefore one alternative is to improvise the user side of the application to maximize user-perceived Quality of Service (QoS).

Some of the existing real-time applications, such as Voice over IP (VoIP) and other multi-media, either use a constant bit-rate utilizing UDP or streaming with TCP connections. UDP is suitable for real-time applications but it does not have any congestion avoidance or flow control scheme. As Floyd and Fall [1] suggest, to ensure fairness among all competing flows it is imperative that every network flow implements some congestion control at the end-user. TCP implements a congestion avoidance policy of linear growth and one-half rate reduction. But this policy is excessively conservative and results in an undesirable jittery steady-state behavior. For this and other reasons TCP is handicapped in dealing with most real-time applications. Moreover, TCP's congestion avoidance policy is a reactive control strategy. It reacts only after congestion has occurred in the network. Real-time applications using schemes that retain the useful features of UDP while at the same time implement some form of predictive control techniques can be expected to greatly improve user-perceived quality. The current research suggests adapting model based predic-

---

This thesis follows the style of *IEEE Transactions on Automatic Control*.



tive algorithms to solve the above problem, given their decade long proven success in the process industries [2].

Process systems and computer networks have three close analogies. Both systems have only a positive actuator force. In a chemical plant, once a chemical/mixture is added to the system, there is no means to remove it. Similarly in a computer network, once a packet is released from the source, there is no way for the source to retrieve it back from the network in order to mitigate the effects of congestion. The packet can only be retrieved by the user at the destination if not lost. The second similarity is the existence of delays in the forward and feed-back loops. In a chemical plant, sensors usually have a reaction time. Moreover, many chemical processes exhibit dead-time in their system. In network based real-time applications, this corresponds to the delay in the forward and feedback paths respectively. A WAN exhibits a dead-time equal to the path round-trip time because the source can not sense the effect of an input till the round-trip. The third similarity is that both systems have many situations where the user has prior knowledge about the near future inputs to the system in terms of the input disturbances. For example, in a chemical plant, for situations occurring in cyclical operations, the operator knows before hand that the reference input is going to change. Similarly, in applications such as media streaming, we know the content that is to be streamed. In certain cases, models such as those proposed by Bhattacharya *et al.* [3] have been developed to predict the input bit-rate. Such analogous models can also be developed for real-time applications such as VoIP. In the process industries, Model Predictive Control (MPC) has been very effective and it is expected that because of the above similarities, MPC can also be successfully extended to networked real-time applications also.

### A. Research Objectives

The first step in developing real-time control for networked applications is to identify the signals that characterize the dynamics of the system. For an end-user, accumulation of bytes in the network is an appropriate signal, when compared to traditional signals such as packet loss or end-to-end delay sequence. Accumulation has three key advantages. First, the accumulation signal reflects both packet loss and the effects of delays in the network. Second, the accumulation signal is a continuous-time signal. It has a definite value at each instant of time. Third, the accumulation signal, when seen as the number of bytes accumulated, can be used in modeling the system under consideration with bit-rate as the input and accumulation as the output.

There are two stages in developing a model predictive control system. The first stage is to develop one or multiple predictors and the second stage is to design one or multiple controllers. The current research revolves around the development of predictors that can capture the end-to-end dynamics of the network from the perspective of a single flow. The predictors developed are analyzed for two key features, the prediction accuracy with increasing prediction horizon and the prediction accuracy with increasing accumulation levels. Reasonably good prediction over increasing prediction horizons gives us more time to implement the control action and also increases the number of degrees of freedom with which the input to the system can be changed. The second feature, namely the accuracy with increasing accumulation levels, indicates the ability of a predictor to capture the congestion in a network: the most important regime where application control must be effective. The user perceived QoS in real-time applications can degrade not only because of packet loss but also due to increase in end-to-end delay, which beyond a threshold is equivalent to a loss. Therefore the research is concluded with an analysis on the interplay between input

bit-rates, packet loss and increase in the end-to-end delay.

## B. Literature Review

### 1. End-to-End Analysis

Innumerable studies have been performed on capturing the characteristics of the Internet. Bolot [4] samples the network using small UDP packets at constant sampling rate. He observes clustering of probe packets and rapid fluctuations of queuing delays over small intervals. The surprising observations is that probe packet losses are essentially random except for the period when the probe traffic uses a large fraction of the available bandwidth. Paxson [5] investigates the end-to-end dynamics using TCP because he believes that TCP constitutes the widely used protocol and hence represents the end-user observations. He observes that the likelihood of a packet loss increases by an order of magnitude if its predecessor was lost. This suggests the existence of a stochastic process rather than a random process governing packet loss. In a more recent study, Bovy *et al.* [6] suggest that 84 % of their measurement paths are typical histograms possessing a Gamma-like shape with subexponential (or heavy) tail. Though the above studies provide a good insight on modeling the stationary aspects of a network, these techniques address only the statistical characteristics of the network and they are not very useful for dynamical studies and control purposes.

### 2. Network Simulators

Simulators such as ns-2 [7] have been developed by researchers to capture the characteristics of the Internet. As our experience demonstrates, simulation scenarios that can be developed on these simulators can not capture the real-world end-to-end behavior of the Internet. Paxson and Floyd [8] suggests that simulation of the Internet

is a very difficult problem because of the wide heterogeneity in link properties, topology, protocol differences and the applications that generate network traffic. The other key factor is that the Internet has been evolving very drastically and it is therefore very difficult to obtain statistical models. Paxon identifies that the key invariants of the problem are the long-term dependencies based on self-similarity and Pareto distribution, with a heavy tail for packet arrival.

### 3. Predictability Analysis

Sang and Li[9] present a predictability analysis of network traffic. They provide an upper bound for the optimal performance of online traffic prediction. They find that prediction deteriorates quickly with increasing prediction horizon. They also suggest that predictability of traffic at the backbone increases due to aggregation and multiplexing of different flows. Shah *et al.* [10] present an analysis on the predictability of data networks. They use simulations from ns-2 and utilize linear and nonlinear AR models along with state-space techniques to compare their prediction to a simple predictor. They suggest that prediction at larger time scales is more promising than at smaller time scales. They also suggest that in the case of TCP connections, the mean packet arrival rate is as good an estimate as the one obtained from dynamic predictors based on Auto-Regressive Moving Average models and state space models.

### 4. System Identification as Applied to the Internet

Ohsaki *et al.* [11], [12] use system identification techniques to model the Internet. They use packet inter-departure time from the source as the input and round-trip time variation as the output. They show that an Autoregressive Exogenous (ARX) model is suitable for Local Area Networks (LAN) and also for a Wide Area Networks

(WAN), if the bottleneck link is shared by a small number of users. You and Chandra [13] suggest that only a subset of TCP packet arrivals exhibit both non-stationary and nonlinear features. The remaining traffic has a good agreement with packet loss predicted using a Threshold Autoregressive (TAR) model.

## 5. Neural Networks and Fuzzy Logic

Parlos [14] suggests using neuro-predictors to perform multi-step-ahead prediction of network delay. He points out that though multi-step prediction may be inaccurate when compared to a single-step prediction, they are more useful in terms of improving QoS. Wang *et al.* [15] use Radial Basis Functions (RBF) and linear predictors to predict network delays. They find that once the RBF network is trained sufficiently, it outperforms the linear predictors. These predictions are used in developing a fuzzy logic-based QoS to be employed in Internet-based tele-operation. Jiang *et al.* [16] develop a “model-free” fuzzy time-series predictor for predicting packet arrival patterns for multi-media traffic. They verify their model using data obtained from a simulated continuous-state autoregressive Markov model along with a ‘Star Wars’ video-traffic data set. Doddi [17] developed auto-regressive and neural networks based predictors and analyzed them for various ns-2 simulation scenarios. Edmund *et al.* [18] use back-propagation based feed-forward neural network to perform time-series prediction of network delays. They find that their neural network is an attractive alternative to traditional regression techniques. But the shortcoming of the work is that the training and test data was obtained using simulation of an AR Markov model.

## 6. Accumulation as a Feedback Signal

Traditionally most of the signals used for deducing the characteristics of the Internet were based on delay or loss signals. Xia *et al.* [19] suggest that congestion control algorithms must be based on accumulation of bytes in the network. They offer a congestion control algorithm based on a bit-by-bit fluid model. This is one of the first few papers that suggest using accumulation as a feedback signal. Khariwal [20] shows that adaptive control can be applied to best effort networks to improve QoS. He develops an autoregressive model to predict the accumulation signal and uses a controller to maintain accumulation at a reference level. The above scheme resulted in a decrease in packet losses with negligible decrease in interactivity. Konstantinou [21] developed an end-to-end fluid model to develop a predictive controller to improve QoS of real-time applications. The key draw back of the above two works is that the results are based on simulated environments. Real-world traces show high variability and sharp spikes in end-to-end delay, as compared to simulated traces.

### C. Proposed Approach

The objective of the present research is to develop end-to-end “black-box” models of single flows in WANs that can be used to predict network characteristics and can aid in the development of control schemes. The network is probed at a constant sampling rate using a constant bit-rate UDP flow consisting of fixed packet size. In this case, accumulation of the number of bytes is equivalent to the accumulation of number of packets at any given instant. In later sets of experiments, the bit-rate is varied to study its effect on delay and packet loss. The lost packets are removed from the accumulation signal at the earliest moment a packet loss is detected.

The predictors are developed initially on ns-2 simulation scenarios and then on

real-world traces obtained from a popular overlay network, called Planet-Lab [22]. To ensure that the traces obtained represent typical Internet users, DSL nodes and International nodes are selected as the end points for the experiments. These traces are then used in training linear and non-linear models. The linear time-series model used in this research is an auto-regressive model. For non-linear models, Sparse Basis Functions [23] and Radial Basis Functions (RBF) are chosen. Sparse Basis Functions (SBF) are selected because of the promising results presented by Atiya *et al.* [23] in predicting the time-series of video source traffic. The RBF networks were selected because they can be easily trained and require very few degrees of freedom. A comparative analysis on the predictor performance is then conducted for increasing prediction horizons and for increasing accumulation levels. The reason for increasing prediction horizon is that to implement effective control, the farther the prediction horizon, the greater the degree of freedom in adapting and shaping the input to the system. The predictors are also analyzed for higher levels of accumulation because the critical and desirable feature of a predictor is its ability to predict congestion.

In real-time applications, interactivity is hampered by two kinds of losses, namely the loss of packets and the arrival of packets after their deadline due to congestion in the network. Therefore, we also analyze the effect of bit-rate on interactivity loss. To study the sensitivity of the network to changes in bit-rate, we switch the bit-rate between a low value and a high value every few seconds. This experiment is expected to reveal the open-loop step-response of the network path under consideration.

#### D. Contributions of the Current Research

Predictive control oriented modelling of WAN is a relatively new and challenging field. The key contributions of the current research are as follows:

- Development of predictors that are suitable in implementing control of real-time applications across a best-effort WAN.
- Comparative analysis of the predictors with emphasis on their performance in the wake of congestion and with increasing prediction horizons.
- Exploration of the effects of encoder bit-rate on interactivity, which in turn depends on packet loss and large end-to-end delays beyond their interactivity dead-line.

#### E. Organization of the Thesis

The thesis has been divided into six chapters. Chapter II outlines various modelling techniques along with a brief description of the linear and non-linear models used in this research. Chapter III describes the experimental set-up used, the simulation scenarios used and the real-world experiments. Prediction results of end-to-end single flow characteristics and a comparative analysis on each of the predictors is presented in Chapter IV. Chapter V presents the effect of encoder bit-rate on packet loss and end-to-end delay along with an analysis of prediction accuracy for different prediction models. Chapter VI gives a summary of the thesis along with conclusions and recommended future work.



## CHAPTER II

## MODELING AND PREDICTION TECHNIQUES

## A. Introduction

The current chapter is organized as follows. The broad categories in modeling techniques are explained first. These are then followed by an overview of the generic procedure for system identification. This is followed by a description of the linear and nonlinear identification methods used in the current research work.

## B. Modeling Categories

Modeling can be classified into three basic categories. They are *White box models*, *Black box models* and *Grey box models*.

1. **White box models:** White box models are derived from first principles using the physical, chemical, etc. laws. All the parameters used in the models can be determined by theoretical modeling. These models do not depend on data and their parameters have a direct first principles interpretation. These models have good extrapolation capability and are highly reliable. The drawback is that they are time consuming and have application only in well understood processes. They also need detailed domain expertise. Application areas include planning, construction, and rather simple processes.
2. **Black box models:** Black box models are solely based on measurement data. Both model structure and parameters are determined from experimental modeling. The advantages of black box models include short development time and little domain expertise. But these models have unreliable extrapolation prop-

erties and provide little understanding of the underlying physical process. This technique is very effective in modeling rather complex processes.

3. **Grey box models:** Grey box models are a combination or compromise between white and black box models. Besides the knowledge from first principles and the information contained in the measurement data other knowledge sources such as qualitative knowledge formulated in rules may also be utilized in grey box models. Typically, the determination of model structure relies strongly on prior knowledge while model parameters are mainly determined by measurement data.

In reality pure white or black box approaches rarely exist. Nothing is black or white, everything is gray. In light gray box models, the model structure may be determined by first principles but models parameters may be estimated from data. In dark grey box models, a neural network may be used but the data acquisition procedure, e.g, design of excitation signals, requires prior knowledge. Usually, if prior knowledge is clearly the dominating factor, one speaks of white box models and if experimental data is the major basis for modeling, one speaks of black box models. The current research uses grey box models such as Auto-regressive methods, Radial-Basis functions and Sparse basis functions in modeling the accumulation signal from the end-to-end delays.

### C. System Identification Procedure

This section examines the major steps that have to be performed for successful system identification. Figure 1 shows the generic procedure for modeling a dynamic system. The complexity and need for prior knowledge decreases from top to bottom.

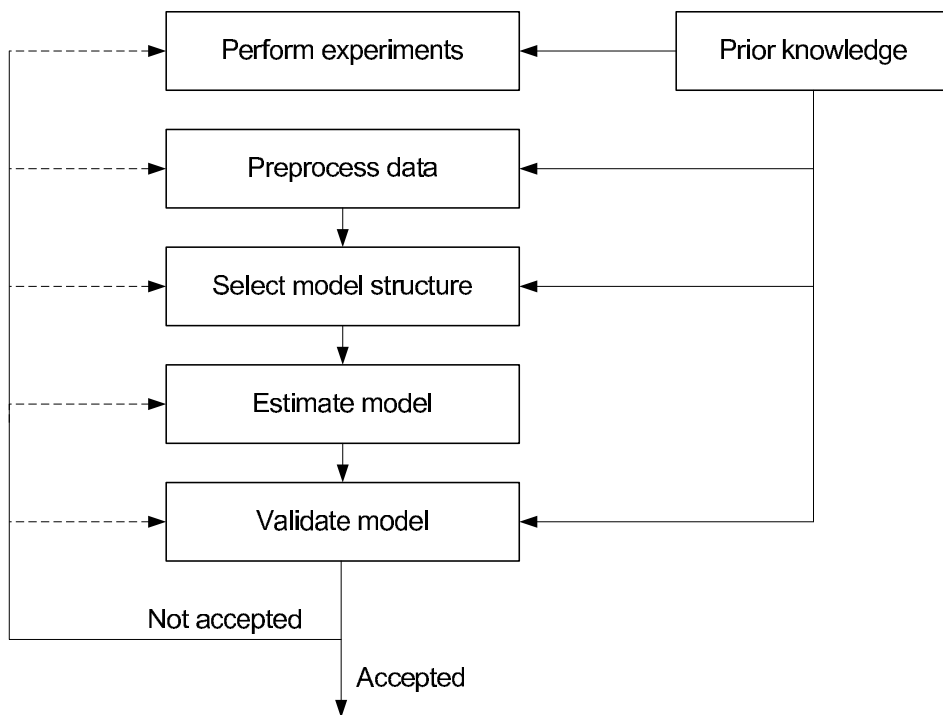


Fig. 1. System identification procedure.

1. **Perform experiments:** This is the first step in identifying the system. It required prior knowledge about the process and the purpose of the model. Deciding which inputs to choose and which outputs represent the dynamics of the system and the experimental setup are the crucial elements of this step. The choice of the output signal is crucial because the process behavior not represented within the data can not be described by the model unless prior knowledge is explicitly incorporated. The excitation signals should ensure that the system is excited at all the operating ranges. In time-series modeling techniques, there are no excitation signals. In such cases, selecting an output data set that is as representative as possible is very critical. Therefore, this step involves significant engineering expertise and prior knowledge.
2. **Preprocess data:** Pre-processing of the data-sets is essential to ensure the training data represents the operating region of interest. This step uses prior knowledge to reduce the complexity of the model and in-turn reduces the modeling errors. Pre-processing may include filtering, de-trending, removal of outliers, detection of steady-state data-sets etc.
3. **Model structure selection:** Selection of the model architecture is highly subjective stage of modeling. Some of the factors influencing structure selection are dimensionality of the problem in terms of the number of inputs and outputs, the constraints on computational complexity, memory, development time and whether the model is to be used online or offline.
4. **Model order and parameter estimation:** This step is carried out by a combination of prior knowledge, trial and error. The overall model complexity is limited by the bias-variance dilemma.

5. **Model validation:** This step checks whether all the previous steps have been carried out successfully or not. The validation criteria is highly problem dependant because in many cases modeling is not the ultimate goal. The model is to be used in designing a controller or a fault detection system. The first step is to check the model on the training data. If the model results in satisfactory performance it is then recommended to test it on a fresh data to evaluate the effects of over-fitting.
6. **Going backwards in the procedure:** The return path from the validation block indicates that System Identification is an iterative procedure. In certain models, the “fiddle” parameters such as the learning rate in a training method, the number of the number of clusters, regressors, neurons, the error threshold that terminates the algorithm etc., need to be manually adjusted and iterated because either the identification algorithm is not sophisticated enough to optimize these parameters automatically or the objective of the optimization function cannot be properly expressed.

#### D. Linear Model Structures

The current research applies parametric Linear System Identifications methods in modeling the accumulation signal. These models attempt to describe the true process behavior exactly with a finite number of parameters. The parameters are determined by optimizing some objective function by methods such as linear regression. Linear models have the advantages that they are computationally easy to develop, and the controller design utilizing them is much simpler. Figure 2 shows the general linear model structure.

Other forms of linear modeling can be thought as being derived from this general

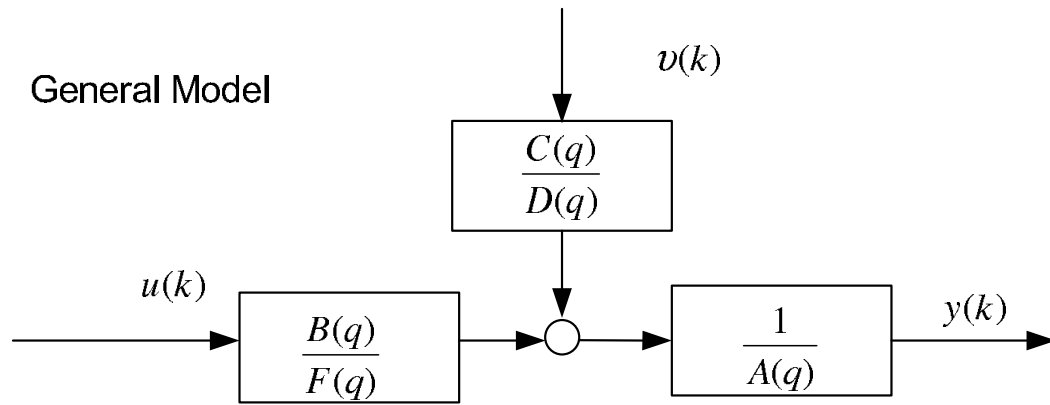


Fig. 2. General linear model.

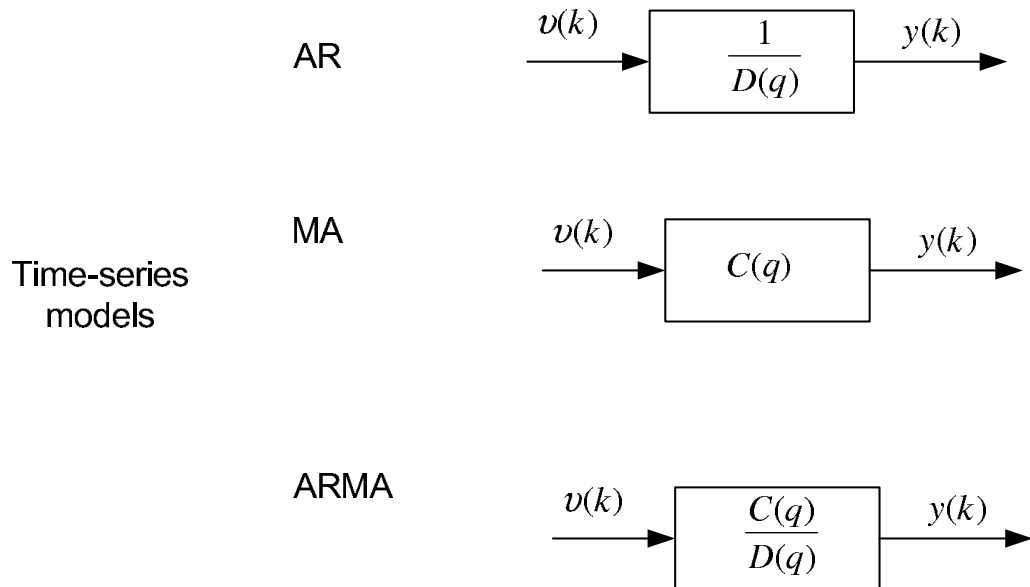


Fig. 3. Linear time-series models.

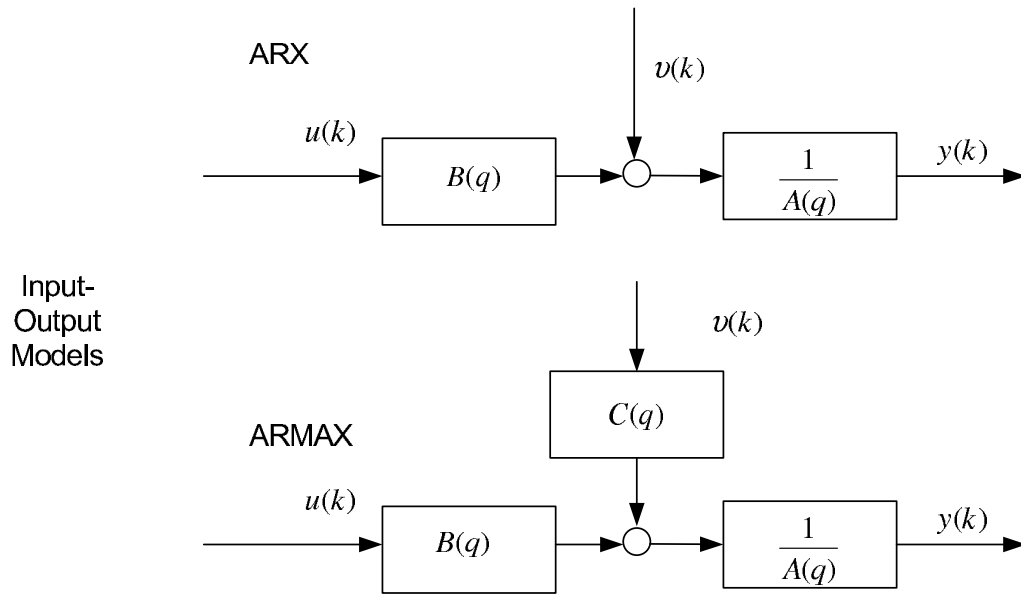


Fig. 4. Linear input-output models.

model. Time-series models do not have any input. In time-series models, the relevant input is hardly known or the number of possibly relevant inputs is huge. Such models have been used effectively in modeling stock prices, currency exchange rates etc. The generic time-series models are depicted in Figure 3. A time-series model with just the denominator polynomial is called the autoregressive (AR) model (see eq. 2.1) .

$$y(k) = \frac{1}{D(q)}\nu(k) \quad (2.1)$$

A time-series model with just the numerator polynomial is called a moving average (MA) model (see eq. 2.2) .

$$y(k) = C(q)\nu(k) \quad (2.2)$$

A time-series model with a numerator and a denominator polynomial is called autoregressive moving average (ARMA) model (see eq. 2.3) .

$$y(k) = \frac{C(q)}{D(q)}\nu(k) \quad (2.3)$$

It is obvious that a model based on time-series without considering any relevant inputs can not be expected to yield accurate results. More accurate models are constructed by incorporating one or more input variables. This input  $u(k)$  is called an *exogenous* input. Figure 4 shows the main input-output models used in this research. The *autoregressive model* is given by equation (2.4) and an autoregressive moving average model is given by equation (2.5). ARMAX is more flexible than ARX owing to the moving average polynomial.

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}\nu(k) \quad (2.4)$$

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}\nu(k) \quad (2.5)$$

### 1. Auto-Regressive and Auto-Regressive with Exogenous Input Models

The Auto-Regressive Exogenous (ARX) is the simplest and the most used model structure in system identification. The general Single Input Single Output (SISO) ARX model can be expressed by the following linear difference equation :

$$\begin{aligned} y(t) = & a_1y(t-1) + \dots + a_{n_y}y(t-n_y) \\ & + b_1u(t-n_k) + \dots + b_{n_u}u(t-n_u-n_k+1) \\ & + e(t) \end{aligned} \quad (2.6)$$

where  $u(t)$  and  $y(t)$  are the input and the output of the SISO ARX model,  $n_y$  and  $n_u$  are the number of past outputs and the number of past inputs used in the model, and  $n_k$  is the pure time delay (the dead time) in the system. The coefficients  $a_1, \dots, a_{n_y}$  and  $b_1, \dots, b_{n_u}$  are known as the model parameters.



Based on ARX model of equation (2.6), the following Single Step Predictor of the system output can be obtained:

$$\hat{y}(t|t-1, \theta) = \phi^T(t)\theta \quad (2.7)$$

where,  $\varphi(t) = [y(t-1), \dots, y(t-n_y), u(t-n_k), \dots, u(t-n_u-n_k+1)]^T$ ,

$\theta = [a_1, \dots, a_{n_y}, b_1, \dots, b_{n_u}]^T$ .

Equation (2.7) is in the form of a linear regression with the model parameter vector  $\theta$  the regression vector. The parameter vector  $\theta$  in the equation (2.7) is estimated using the least-squares method for the parameter vector  $\theta$ , minimizing the mean-square of the prediction error.

The Auto-Regressive (AR) model is a special case of the ARX model where only past values of the output are used for modelling the system. The AR model is a time-series model.

## 2. Auto-Regressive Moving Average with Exogenous Input Model

A more general input-output model is given by the Auto-Regressive Moving Average Exogenous Model (ARMAX). The AR in the ARMAX model refers to the autoregressive part, and the MA is the moving average and X corresponds to the extra input called the exogenous variable. The ARMA model formulates the disturbance term as a moving average of a white noise process. The Single Input Single Output (SISO) ARMAX model can be represented by the following equation:

$$\begin{aligned} y(t) = & a_1 y(t-1) + \dots + a_{n_y} y(t-n_y) + \\ & b_1 u(t-n_k) + \dots + b_{n_u} u(t-n_u-n_k+1) + e(t) + \\ & c_1 e(t-1) + \dots + c_{n_e} e(t-n_e) \end{aligned} \quad (2.8)$$

where,  $n_e$  is the number of past noise terms used in the model,  $e(t)$  is a white

noise process, and the other variables are the same as in the ARX model.

The ARMAX predictor can be written as a scalar product between the data vector  $\varphi(t; \theta)$  and the parameter vector  $\theta$ :

$$\hat{\mathbf{y}}(t|t-1, \theta) = \phi^T(t, \theta) \quad (2.9)$$

where,  $\varphi(t; \theta) = [y(t-1), \dots, y(t-n_y), u(t-n_k), \dots, u(t-n_u-n_k+1), e(t-1, \theta), e(t-2, \theta), \dots, e(t-n_e, \theta)]^T$ , and  $\theta = [a_1, \dots, a_{n_y}, b_1, \dots, b_{n_u}, c_1, c_2, \dots, c_{n_e}]^T$ .

The model dependency was indicated by including  $\theta$  as an argument to  $\phi$  in equation (2.9). The equation (2.9) is in the form of a pseudo-linear regression and hence the least squares method can be used to solve for  $\theta$ . The Auto-Regressive Moving Average (ARMA) model is a special case of the ARMAX model, where no input or exogenous variable is used while modelling the system.

The system identification toolbox provided by The MathWorks, Inc., is used for estimating the parameters of the above linear models.

## E. Non-Linear Model Structures

### 1. Radial Basis Functions (RBF)

Most real-world processes exhibit non-linear behavior. Radial basis functions are very effective in capturing the non-linear behavior of systems. This section describes the formulation of the RBF networks. Figure 5 shows a neuron in an RBF network and the structure of an RBF network. The distance of the input vector  $\underline{u} = [u_1 u_2 \dots u_p]^T$  to the center vector  $\underline{u}_i = [c_{i1} c_{i2} \dots c_{ip}]^T$  is calculated using the norm matrix  $\underline{\Sigma}_i$  as the weighting function. This distance  $x$  is then transformed by nonlinear activation function  $g(x)$  given in equation (2.11). Several such neurons are used in parallel and are connected to an output neuron. The final output  $\hat{y}$  is a weighted output given by

equation (2.12).

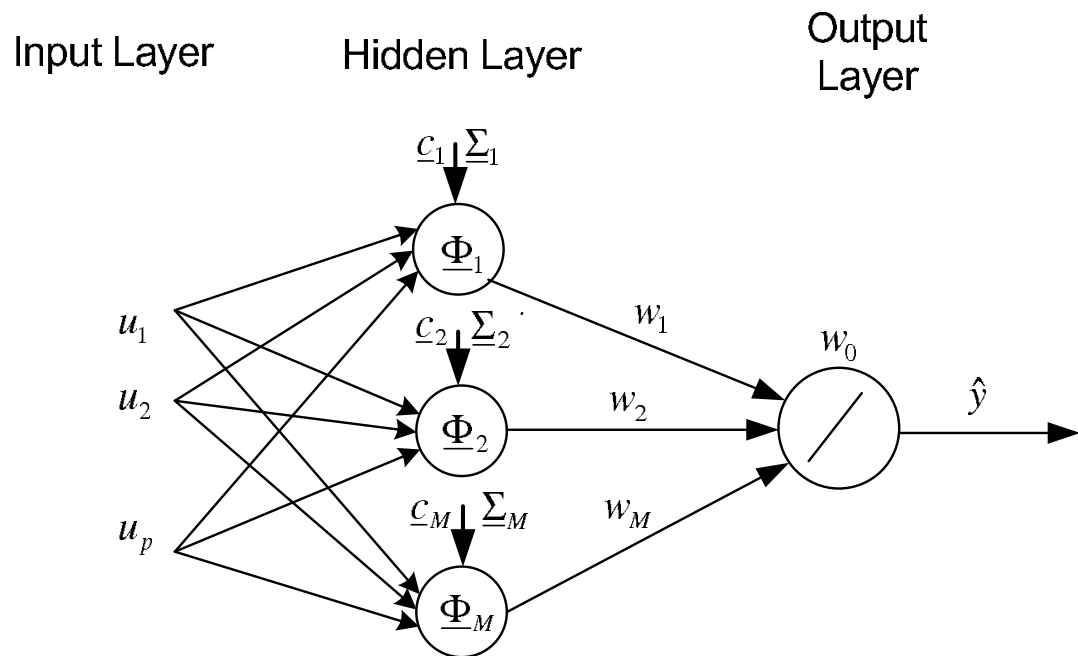
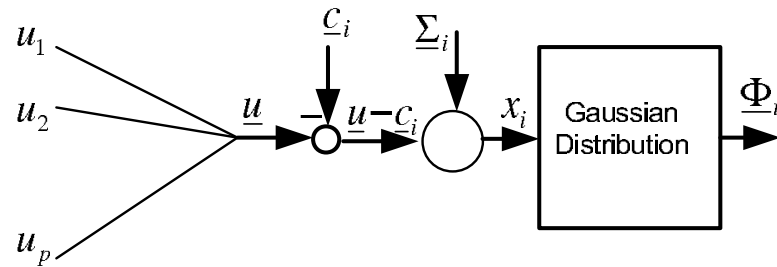


Fig. 5. Schematic diagram of RBF networks.

$$x_i = \|\underline{u} - \underline{c}_i\|_{\underline{\Sigma}_i}^2 \quad (2.10)$$

$$g(x) = e^{-\frac{1}{2}x^2} \quad (2.11)$$

$$\hat{y} = \sum_{i=0}^M w_i \phi_i(\|\underline{u} - \underline{c}_i\|_{\underline{\Sigma}_i}) \quad (2.12)$$

The norm matrix  $\underline{\Sigma}_i$  is usually chosen to be diagonal. In such a case, the total number of parameters becomes  $2Mp + M + 1$  where  $M$  is the number of hidden layer neurons and  $p$  is the number of inputs.

The Radial Basis Functions toolbox of MATLAB has been used to train the network. The input to the system,  $\underline{u} = [u_1 u_2 \dots u_p]^T$ , consists of all the segments of the accumulation signal of the form  $\underline{u} = [a(k) a(k-1) \dots a(k-p)]^T$  where  $a(k)$  is the accumulation series and  $p$  is the order of the RBF network.

## 2. Sparse Basis Functions

In this research, Sparse Basis Functions (SBF) are used based on formulations of [23]. This technique is based on constructing a very large set of possible inputs (“the basis”). These basis functions can be linear combination of inputs such as moving averages, first or second difference. Nonlinear basis functions such as exponentials, square-roots and product of the past inputs have also been included. The prediction is based on a linear combination of a few selected inputs. However the selection of the inputs that are combined is adaptive and varies dynamically from one time sample to the next. An algorithm is given in [23] for the adaptive selection of inputs and recursive update of weights. For single-step prediction, the computation is carried out recursively. For multi-step prediction, a fixed number of lags are used to choose the best basis functions. The key parameters for the single-step predictor are the forgetting factor and the initialization data points. The forgetting factor needs to

be high enough so that the predictor has memory of the dominant dynamics of the system. The initialization data points ensure that the selected basis does not result in an unstable set.

#### F. Chapter Summary

This chapter gives a brief description of the linear and non-linear methods used for modelling. The linear tools are simple and effective for the linear systems, but cannot capture the non-linearity in most of the real-world phenomenon. This chapter also introduces radial basis functions along with sparse basis functions for non-linear empirical modeling.

## CHAPTER III

### TIME-SERIES PREDICTION OF ACCUMULATION SIGNAL

#### A. Introduction

The current research aims at ‘black-box’ modeling of the end-to-end characteristics of Wide-Area-Networks (WANs). ‘Black-box’ models depend heavily on data collected from simulations and real-world experiments. This chapter explains the experimental setups used in collecting the data to train the empirical models, followed by a discussion on development of the linear and non-linear predictors. The chapter finally concludes with a comparative analysis on the performance of the various predictors developed.

##### 1. Experimental Set-up

Figure 6 shows the generic experimental set-up used in this research. Every experiment consists of a source application and a destination application. The source application sends a packet to the source transport layer which in turn injects it as a UDP packet into the network. In all the experiments we select constant inter-departure time for the packet trains. These packets then travel through the Internet and reach the destination transport layer. The destination transport layer unwraps the UDP header and sends them to the receiver application.

##### 2. Ns-2 Simulation Setup

Figure 7 shows the ns-2 simulation set-up. The simulation consists of a source and a destination node connected via a single bottle-neck link consisting of 10 mbps with a propagation delay of 30 ms. The cross traffic is generated by 230 TCP nodes and 10

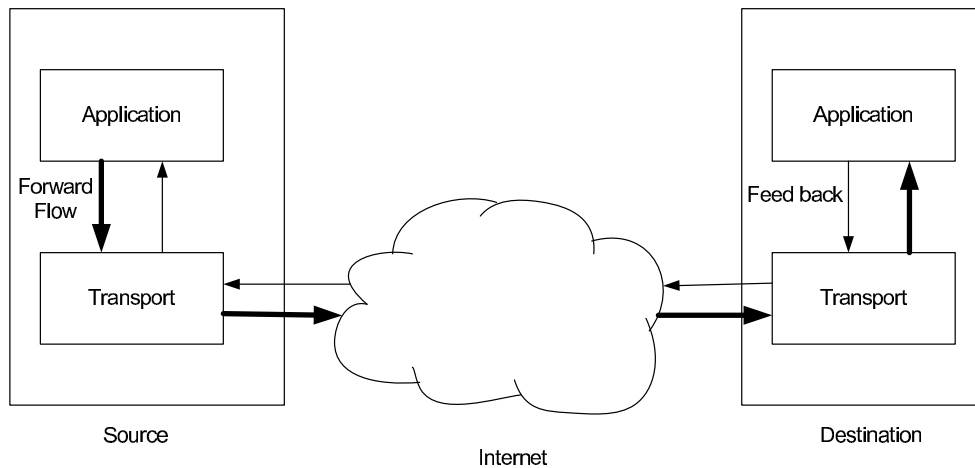


Fig. 6. General experimental set-up.

UDP nodes. The percentage of the probing flow is restricted to 1% of the total traffic on the congested link. This ensures that the probing traffic does not influence the network characteristics. The ratio of TCP to UDP traffic is chosen such that TCP traffic is about 90% of the total traffic. The TCP traffic is again generated such that 90% of it consists of HyperText Transfer Protocol (HTTP) traffic. The above mixture of traffic ensures that the cross-traffic is representative of a real-world network. All the links are chosen to be duplex and drop-tail.

### 3. Real-world Network Experiments

Real world network experiments were conducted on PlanetLab. PlanetLab is an overlay network spread across the world with close to 569 machines, hosted by 270 sites, spanning over 25 countries. On each PlanetLab node, the user creates a ‘slice’. Each slice is programmed such that it is independent of all other slices on the node. At the same time, the node owner has control over all the slices hosted on that node. The node owner can place restrictions such as the maximum bandwidth used by each

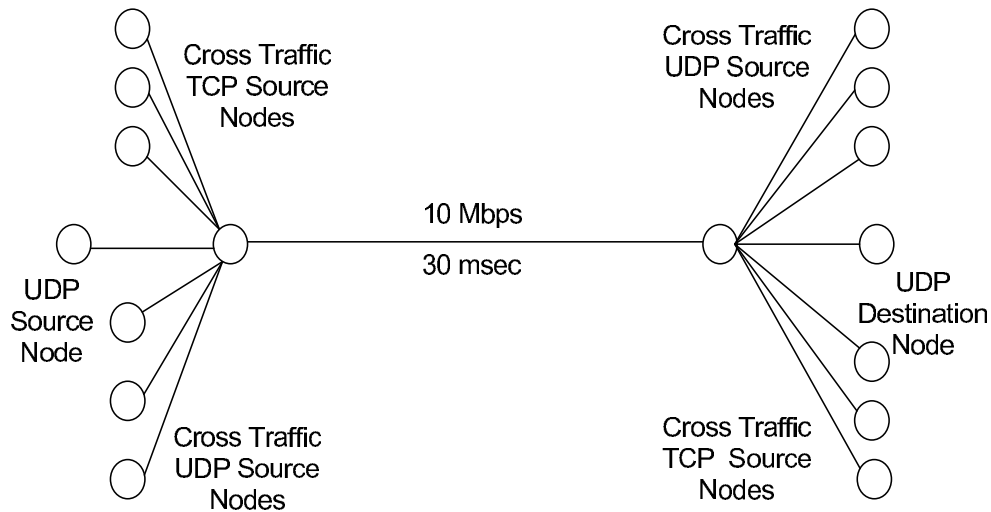


Fig. 7. ns-2 simulation set-up.

slice. The owner also has control on the hardware architecture used by the slice. The key advantage of PlanetLab is that it provides a test-bed to implement new services on a large scale and test them under real-world network conditions.

#### a. Experiment Description

The experimental set-up used in this research is very similar to the one described in Figure 6. In this case, the source and the destination are nodes on the PlanetLab. There are two differences between these experiments and ns-2 experiments. The first difference is that because the two planetlab nodes are on different computers, the two machines may not be time synchronized. Therefore, to calculate the one-way delay, the packets have to carry the time-stamps at the source and at the destination. The packets from the destination are the bounced back to the source. Upon arrival, the packets are again time-stamped. This facilitates calculating the Round-Trip-Time (RTT) for each packet. The second difference is that one must assume that the



minimum  $RTT/2$  calculated for all the packets is the propagation delay. The two time-stamps can then be used to remove the effects of clock-skew while calculating the one-way delay.

## B. Accumulation Signal

The Accumulation signal for a source-destination pair is defined as the difference between the number of bytes introduced into the network by a source and the number of bytes received by a destination. This signal is affected by both the cross-traffic and the input bit-rate. The effect of the input bit-rate is as follows. The accumulation signal is a function of the end-to-end delay experienced by all the packets introduced to the network. The end-to-end delay of each packet is made up of three components: propagation delay, transmission delay and queuing delay. The higher the delay due to any of the above components, the higher is the accumulation signal. Propagation delay is fixed for a given path. Transmission delay is affected by the size of the packets being sent by the source. Packet size is directly proportional to the input bit-rate because the current research uses constant inter-departure time for injecting packets into the network and sampling it. Therefore, the input bit-rate affects transmission delay and hence the accumulation signal. The last but most important component of end-to-end delay is the queuing delay. Queuing delay is affected by cross-traffic across the path selected for the experiment. This cross-traffic may be composed of TCP flows or UDP flows. The applications generating the cross-traffic can be file transfer applications, multi-media applications etc. The relevant inputs for predicting cross-traffic are difficult to identify and are too many to account for. Therefore, the queuing delay (and hence the accumulation signal) is affected by numerous unmodeled factors. Time-series techniques are one of the best options in modeling such complex systems.

This chapter presents the results for time-series prediction of the accumulation signal using linear models such as auto-regressive tools and also with nonlinear models based on Radial Basis Functions (RBF) and Sparse Basis Functions (SBF).

In calculating the accumulation signal, lost packets need special attention. For any time-series prediction technique to be most effective, the signal has to be stationary. Since the accumulation signal is defined as the difference between the bytes sent from the source and the bytes received at the destination, a packet that is lost results in a permanent increase in the accumulation signal by a value equal to the size of the packet. In the case of constant packet size, accumulation signal is equivalent to packet accumulation. Here, the accumulation signal will increase by a packet for each packet that is lost. Therefore the accumulation signal has to be de-trended. The current research detrends the signal based on the assumption that there is no packet reordering. Experimentally, reordering was found in less than 0.1% of all the packets for the paths that were tested for this experiment and hence the assumption is reasonable. De-trending the accumulation signal is performed as follows. Suppose that at a given instant, the destination receives a packet with a sequence number 'k' and the next packet it receives is with sequence number 'k+n'. Because of the assumption of no reordering, it can be concluded that packets with sequence numbers 'k+1', 'k+2'.... 'k+n-1' have been lost. The accumulation due to these packets is removed from the accumulation signal at this instant. This technique ensures that the accumulation signal closely reflects the congestion level in the network with minimal time lag.

### C. Error Calculation Technique

Since the target application to be deployed is a real-time media application over the Internet, the most important period where prediction is required is during congestion.

A congestion period is when the packets being injected into the network are not being extracted out at the destination. This directly corresponds to a rise in accumulation signal. Moreover, if the prediction accuracy is good over longer prediction horizons, the control algorithm can avail more degrees of freedom to control the application. Therefore, the most important features to be analyzed in developing a predictor are its performance with increasing predicting horizons and for high accumulation levels. If the Noise-to-Signal ( $NSR$ ) ration is used in comparing the predictors, one can not analyze the predictor performance for the above two features. Therefore the predictors are analyzed using a new technique described below. Most of the plots use three dimensions, one for the prediction horizon, one for the accumulation level and the third for the relative error. Each point on the graph  $G(A, k)$  is the ratio  $e_{p1}(A, k)/e_{p2}(A, k)$  of the two predictors,  $p_1$  and  $p_2$ , under consideration. The following equation is used to determine  $e_p(A, k)$

$$e_p(A, k) = \sum_{i=1}^{i=N} (\hat{y}_p(i+k) - y(i+k))^2 \quad (3.1)$$

$$if(\min(y(i+k), \hat{y}_{SP}(i+k), \hat{y}_p(i+k)) > A)$$

where,  $A$  is the minimum accumulation level beyond which the signal is considered for calculating the error,  $k$  is the number of steps in the prediction horizon,  $y(i+k)$  is the original time series,  $\hat{y}_p(i+k)$  is the  $k$ -step-ahead prediction from predictor ‘ $p$ ’ and  $\hat{y}_{SP}(i+k)$  is the prediction when using the Simple Predictor (described in Chapter II).

#### D. Experiment and Model Specifications

The models developed in this chapter are trained and tested from the data obtained from ns-2 simulations and from two real-world network experiments. Table I shows the details of these experiments. Figure 7 shows the ns-2 simulation setup. The

real-world experiments involve two international paths, both starting from a node in the USA, one ending in IIIT, India and the other in Germany. The bit-rate was maintained at a very low value of 12.48 kbps for ns-2 simulation and path-1 and 6.24 kbps for path-2. Since the DSL lines chosen for these paths were 512 kbps lines, the effect of probe bit-rate on the network cross-traffic can be assumed to be negligible. This bit-rate also matches the low end VoIP bit-rate (eg. ITU G.726 and ILBC-Internet Low Bitrate Codec). Table II shows the predictors that were developed. For each of the predictors, the model parameters were determined based on 4000 samples of training data and 4000 samples of validation data. The prediction results of these predictors was then calculated on a separate set of 4000 samples. A detailed discussion on the development of these predictors is presented in the subsequent sections.

#### E. Autoregressive (AR) Models

This section describes how the auto-regressive (AR) models were developed for time-series prediction and how these predictors perform with increasing prediction horizon and with increasing accumulation levels.

##### 1. Parameter Selection

The first estimate of the AR model is obtained from the autocorrelation plot of the time-series signal. Figures 8, 9 and 10 show the Auto-Correlation Function (ACF) of the accumulation signal for the simulated and actual network traffic for the simulated and actual network traffic. It is to be noted that for the ns-2, the ACF plot is much wider than the real-world data. This indicates that ns-2 simulation does not adequately represent the real-world conditions. In training Autoregressive models, the first approximation of the order is chosen to be the number of lags where the

Table I. Description of Constant Bit-Rate Experiments

<b>Name</b>	<b>ns-2</b>	<b>Path-1</b>	<b>Path-2</b>
<b>Source</b>	node1	planetlab1.gti- dsl.nodes.planet-lab.org	planetlab1.ucb- dsl.nodes.planet-lab.org
<b>Destination</b>	node2	planetlab1.iiit.ac.in	planetlab1.informatik. uni-kl.de
<b>Packet Size</b> (bytes)	50	50	50
<b>Packet inter- departure time (ms)</b>	50	50	100
<b>Accumulation sampling in- terval (ms)</b>	50	50	50

Table II. Model Parameters for Time-Series Prediction

Specifications	ns-2	Path-1	Path-2	
<b>AR</b>				
order	50 (2.5 sec)	40 (1.5 sec)	350 (17.5 sec)	
<b>RBF</b>			<b>RBF1</b>	<b>RBF2</b>
Neurons	60	60	60	32
Radius	20	6	3	64
Order	10 (0.5 sec)	10 (0.5 sec)	10 (0.5 sec)	10+40 at 290 lags (2.5 sec)
<b>SBF</b>				
Initialization samples	400 (15 sec)	400 (15 sec)	400 (15 sec)	
No. of Basis Functions				
(I) Tested	60	60	60	
(II) Selected	5	5	5	
Forgetting Factor	0.99	0.99	0.99	

ACF drops below 0.2. In the ns-2 scenario, this corresponds to 60 lags. In the real-world scenarios, this corresponds to between 30 to 40 lags. But in the case of ‘path-2’, the ACF goes beyond 0.2 at 300 lags and drops down at 350 lags. Therefore, an AR model of order 60 is chosen for the simulated data, an order of 40 for path-1 and an order of 350 for path-2. A second issue to be noted from Figures 8, 9 and 10 is that for ns-2 simulated data, the ACF crosses zero at about 150 lags and thereafter maintains a very low correlation value. But in the case of real-world scenarios, the ACF has a value close to 0.1 even after 300 lags and does not seem to have a downward trend. This is indicative of long term dependencies in the data and suggests that the ns-2 simulations are not capable of capturing all the features of real-world network traffic. Akaike Information Criterion (AIC) is one of the popular methods for determining the model of auto-regressive models. Figures 11 and 12 show the AIC values for the different AR model orders for path-1 and path-2. This also supports the choice of the AR models suggested above. For path-2, if we are to use only the AIC, the suggested model would have been of order between 40 and 60. But to take advantage of the autocorrelation peak between 270 and 350 lags, we use 350 as the AR order.

## 2. AR and SP Prediction Results

This section presents the results of Auto-Regressive (AR) models when compared to Simple Predictor (SP) . Figures 13, 14 and 15 show the prediction accuracy of the AR, RBF and SBF prediction. In this section, discussion will be confined to the AR and SP predictors. Discussion on RBF and SBF predictors is presented in later sections.

From Figure 13 it can be noticed that the *NSR* of a SP increases almost linearly till about 0.4 seconds and then remains steady at 0.2. In the case of path-1 and path-2, the *NSR* does not settle down even after 1 second. Moreover, figures 14

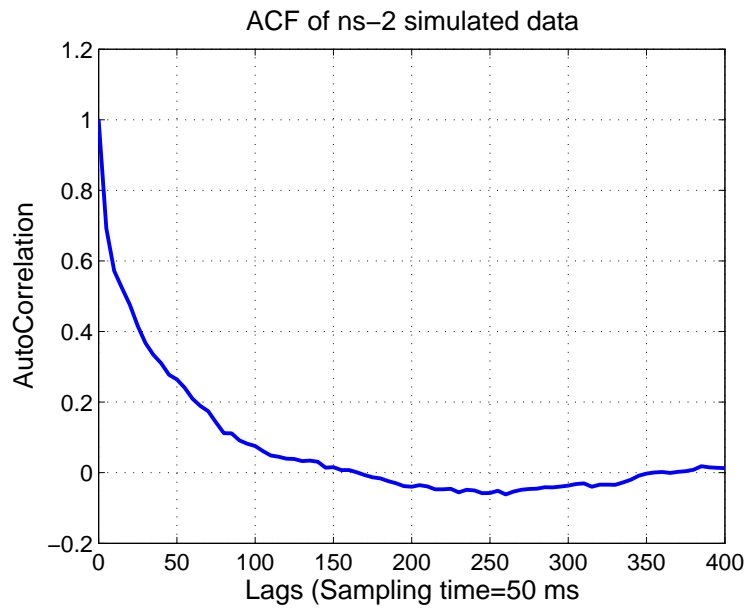


Fig. 8. Auto-correlation functions for ns-2 accumulation data.

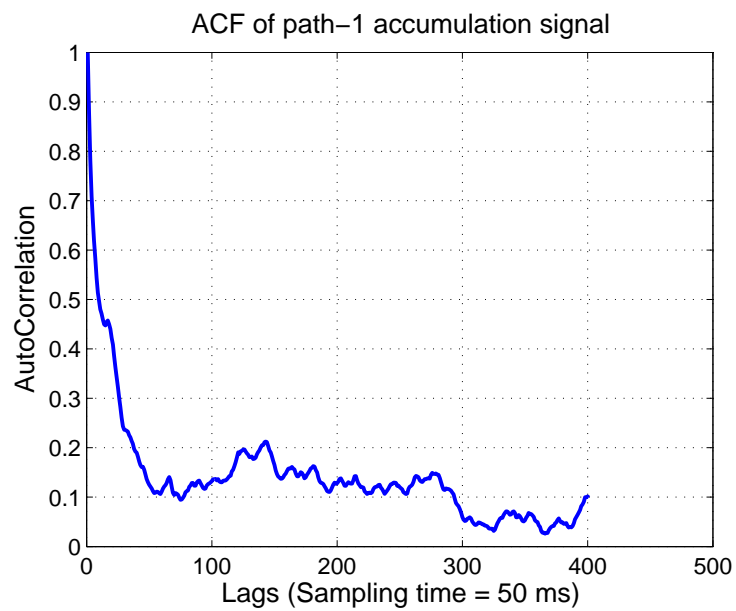


Fig. 9. Auto-correlation functions for path-1 accumulation signal.



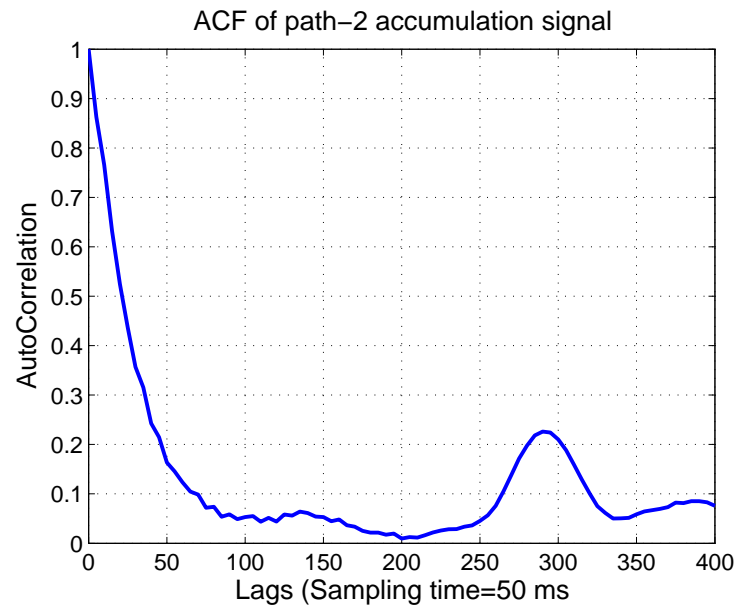


Fig. 10. Auto-correlation functions for path-2 accumulation signal.

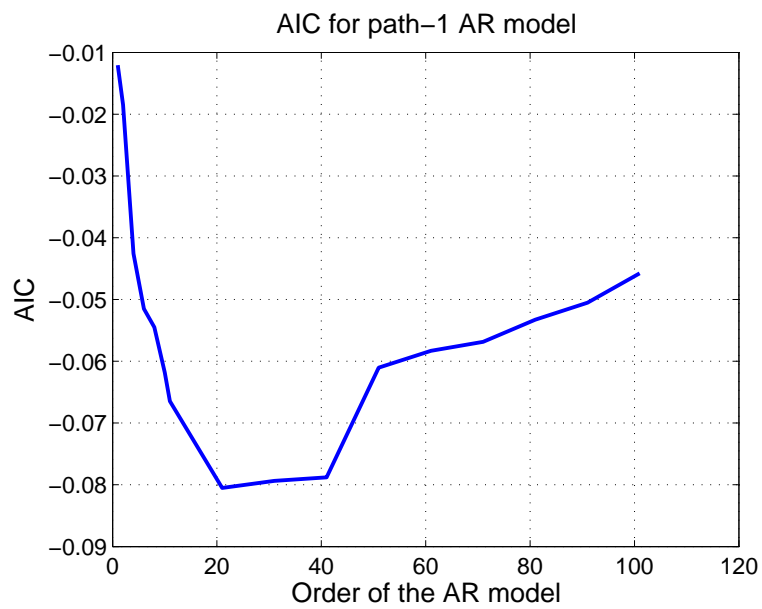


Fig. 11. AIC criteria for path-1 accumulation signal.

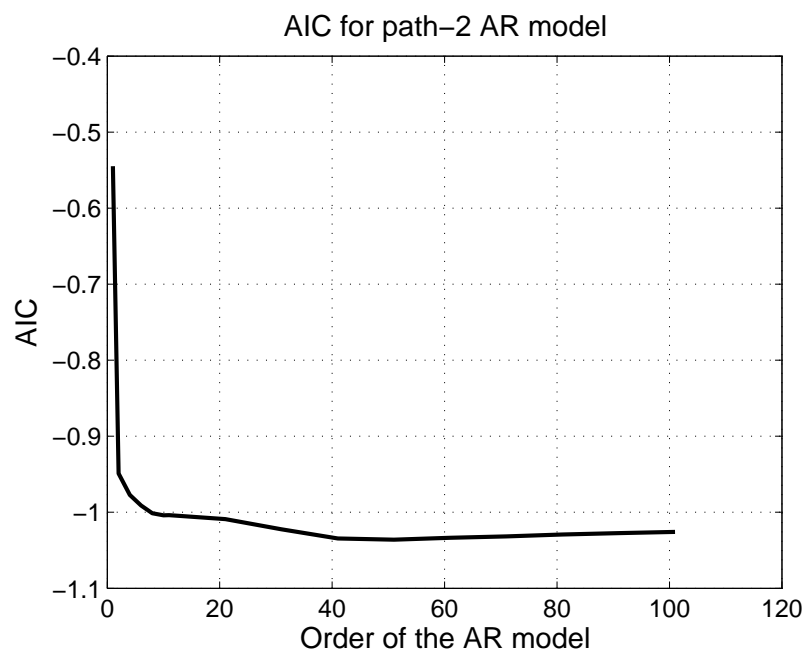


Fig. 12. AIC criteria for path-2 accumulation signal.

and 15 show that  $NSR$  of SP on path-1 and path-2 is very similar. In both cases,  $NSR$  reaches 0.2 at one second prediction horizon. This indicates a slight difference between ns-2 simulations and the real-world network experiments. In all the three cases, it can be noticed that the AR model reduces the  $NSR$  to about 80% of SP.

Figures 16, 17 and 18 show the relative performance of AR vs SP on a fresh data for increasing prediction horizons and higher accumulation levels. The key similarity among these three figures is that at small prediction horizons, AR and SP perform almost equivalently. The second observation is that in all the cases, the prediction for AR and SP does not degrade for prediction horizons beyond 0.5 seconds. In the case of path-2, there seems to be a certain advantage in using AR to predict higher accumulation levels.

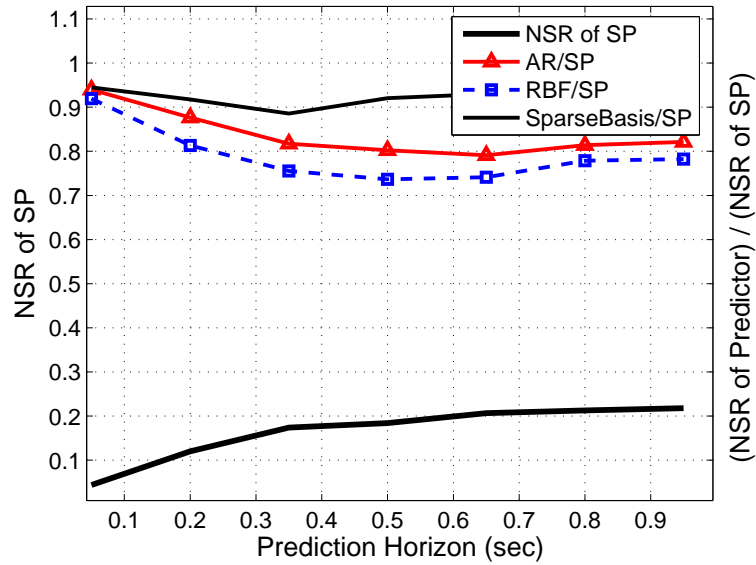


Fig. 13. Comparison of AR, RBF and SBF predictors for accumulation signal on path-1.

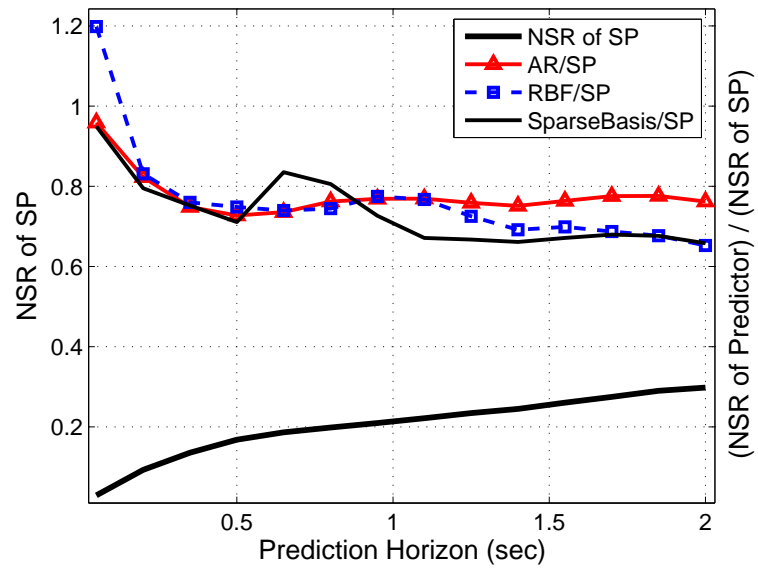


Fig. 14. Comparison of AR, RBF and SBF predictors for accumulation signal on path-1.

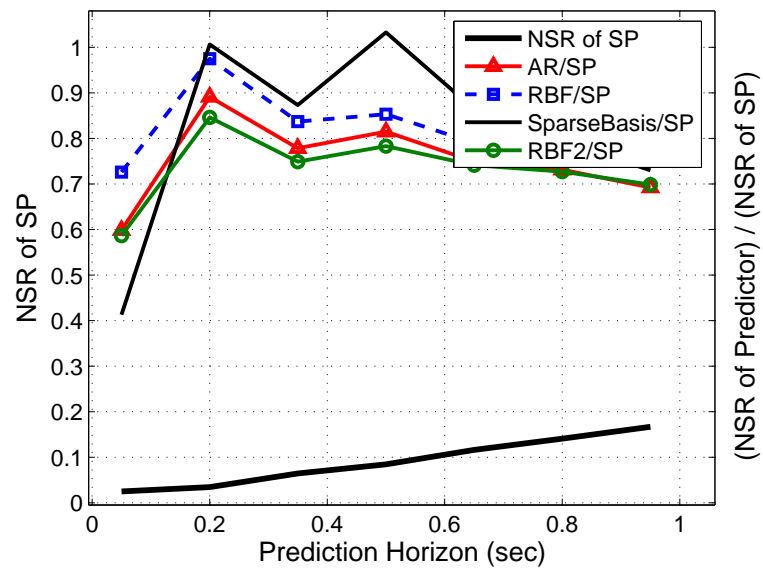


Fig. 15. Comparison of AR, RBF1, RBF2 and SBF predictors for accumulation signal on path-2.

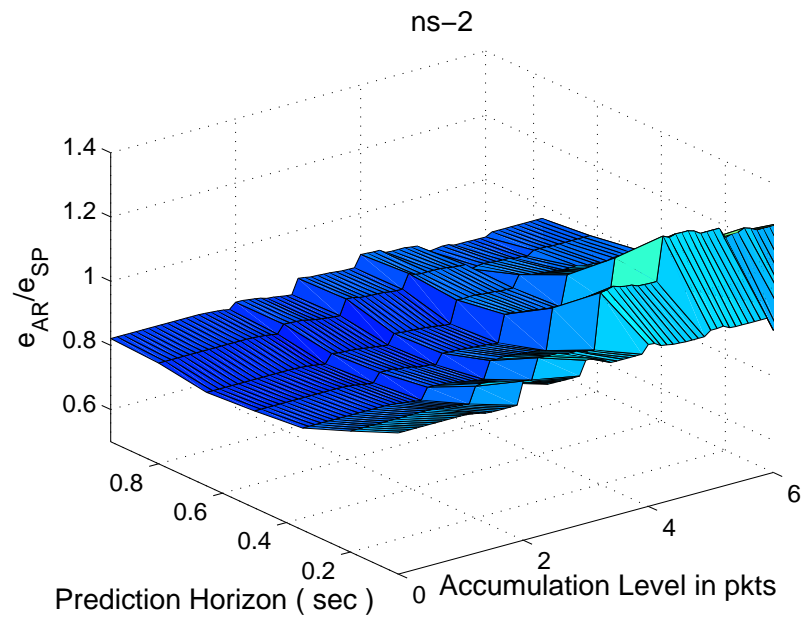


Fig. 16. Comparison of AR and SP for ns-2 accumulation signal.

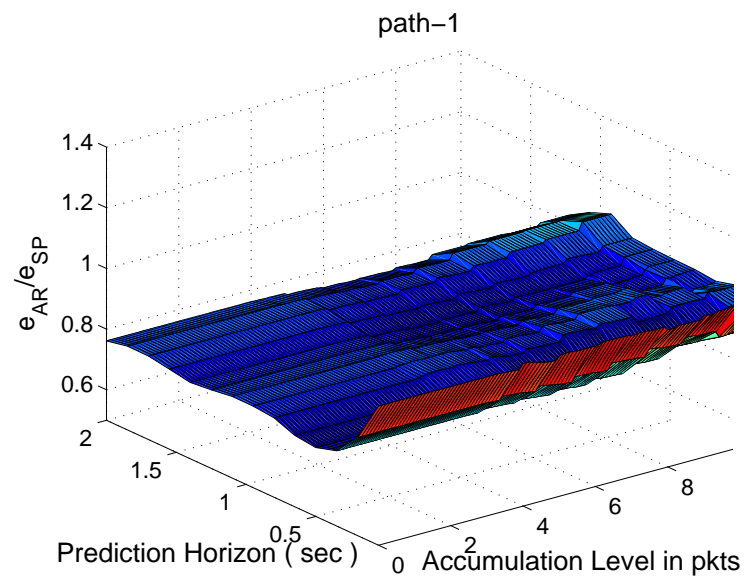


Fig. 17. Comparison of AR and SP for path-1 accumulation signal.

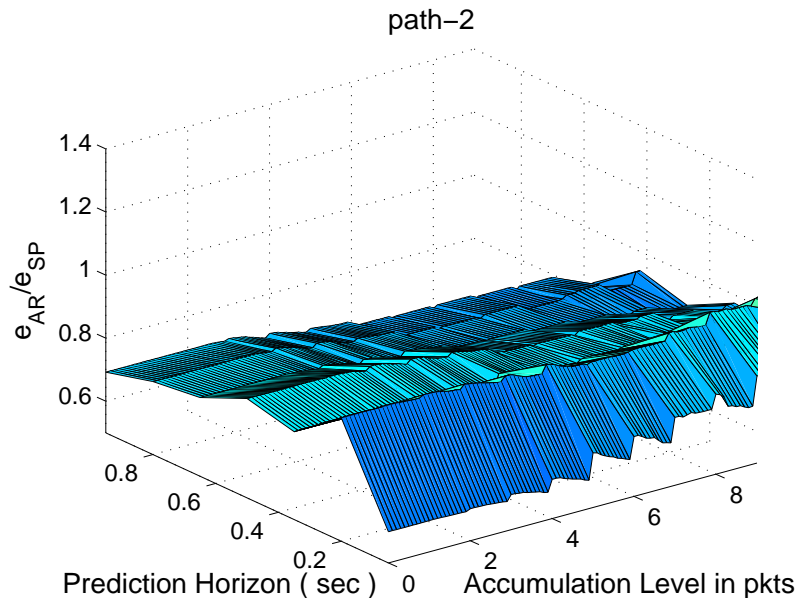


Fig. 18. Comparison of AR and SP for path-2 accumulation signal.

## F. Radial Basis Function (RBF) Models

This section presents results from the application of Radial Basis Functions (RBF) when applied for time-series prediction of packet accumulation. First the tuning of parameters is explained and then a comparative analysis with AR predictor is presented.

### 1. Parameter Selection

The key parameters to be chosen in training RBF networks are the number of neurons and the radius of the Gaussian weight function. A first approximation for the order of magnitude of the radius of the Gaussian weight function can be obtained from the mean geometric distance between the input vectors. For the input vectors of path-1, the mean geometric distance, calculated by the norm-2, is approximately equal to

1.8. Figure 19 shows that 60 neurons and a radius of about 6 give the best prediction accuracy. It can also be noticed that as the radius of the Gaussian weight function increases from 1 to 4, the prediction accuracy improves drastically. Beyond 6, the prediction degrades but it is much more gradual. In the case of path-2, the auto-correlation plot in Figure 10 shows a peak at 290 lags. Therefore, two RBF models were investigated. The first model uses only 10 lags (as in the case of path-1) as is henceforth referred to as RBF1. The second model uses 40 inputs with 270 to 310 lags in addition to the immediate 10 lags. The second model is henceforth referred to as RBF2. These two models are also developed using the same procedure as described for modeling path-1.

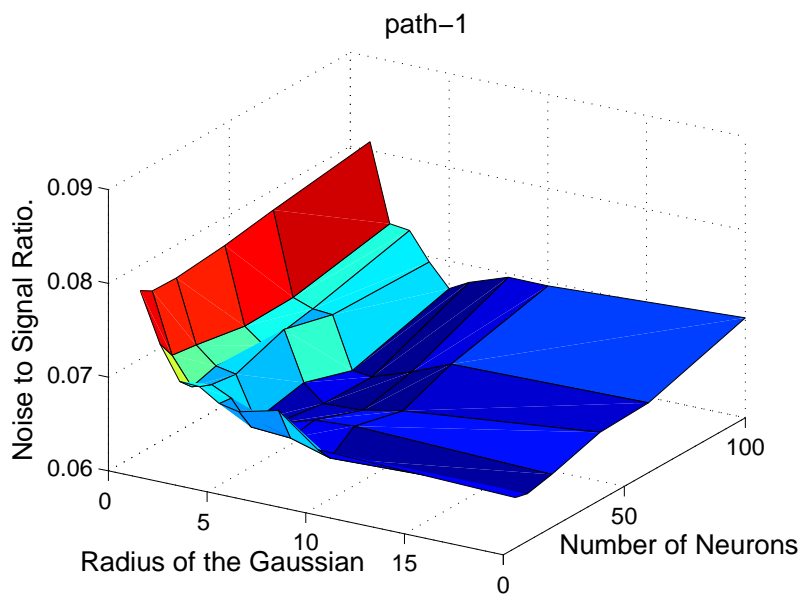


Fig. 19. Parameter selection for RBF network, 0.5 sec prediction NSR for different number of neurons and for different radii for the Gaussian function.



## 2. Radial Basis Function Prediction Results

This subsection presents the of RBF networks as compared to AR predictions. Figures 13, 14 and 15 show the prediction accuracy of the AR, RBF and SBF prediction. It can be noticed that in all the three cases,  $NSR$  of RBF is slightly lower than AR. In the case of ns-2 simulations, RBF consistently resulted in at least about 5% reduction in  $NSR$  when compared to AR at all prediction horizons. In the case of path-1, the improvement of RBF is evident only after 1 second. In the case of path-2, two models of RBF were developed: RBF1 and RBF2. RBF2 used 40 more inputs with 290 lags because of the peak in the auto-correlation plot at 290 lags. RBF1 does not utilize this additional information and therefore resulted in a performance worse than AR. If  $NSR$  were to be the only criterion, RBF2 is the best predictor. But as will be seen in the following discussion, the predictor with the least  $NSR$  is not always the best predictor.

The prediction accuracy of RBF predictors with increasing accumulation levels is discussed below. Figure 20 presents the results of RBF networks for a fresh ns-2 simulated data set. In this case, we find that the RBF network has negligible improvement in the performance when compared with AR. At accumulation levels greater than 5, the performance of RBF starts to improve. Figure 21 shows the performance of RBF networks on path-1. In this case, the RBF overall error is about 80% of AR predictor. The real advantage of RBF networks is at accumulation levels greater than 6 packets. The other key feature to be noticed is that at low prediction horizons, the RBF network is worse than AR. Figure 22 shows the comparison of the RBF network on path-2. The key point to note here is that the AR model uses 350 lags (17.5 sec). But the RBF1 network uses only 10 lags (0.5 sec) to give a comparable prediction. Figure 23 shows the comparison of RBF for the same path with an AR

model of order 40. It can be clearly noticed that the RBF network out-performs the AR model. Figure 24 shows the comparison of RBF2 and AR models. It can be noticed here, that the RBF2 prediction at higher accumulation levels is worse than AR. Thus, even though RBF2 results in a slightly better *NSR*, AR model of order 350 is the best choice. If high computational complexity involved in developing this AR model is a hinderance, an RBF model with just 10 lags can give a comparable performance.

Figure 25 shows 0.5 second prediction of AR and RBF predictors for accumulation signal on an ns-2 simulation. It can be noticed that the RBF predictor is able to predict the peaks in the accumulation signal better than the AR predictor. At lower accumulation levels, the prediction of the two predictors is quite similar. Figure 26 shows 0.5 second prediction of AR and RBF predictors for accumulation signal on path-1. As discussed in the preceding paragraphs, the RBF prediction is only slightly better than AR prediction. Figure 27 shows the prediction of RBF1 and AR predictors on path-2. It can be noticed that the two predictors are quite similar. It is to be noted that the AR predictor uses 17.5 seconds of past data where as RBF used only 0.5 seconds of data. Figure 28 and 29 show the prediction of of RBF2 and AR predictors for two different segments of the accumulation signal. It can be noticed that in the first segment RBF2 predictor is more accurate than AR and is also smoother than AR. But in the second segment, the RBF2 predictor fails to predict certain peaks in the accumulation signal. Thus, RBF2 predictor was good at some segments of the signal and worse at some other parts. Though the *NSR* of RBF2 is less than that of AR predictor, its prediction at certain instants of high accumulation (congestion) is not good. Therefore, the AR predictor is more suitable for path-2.

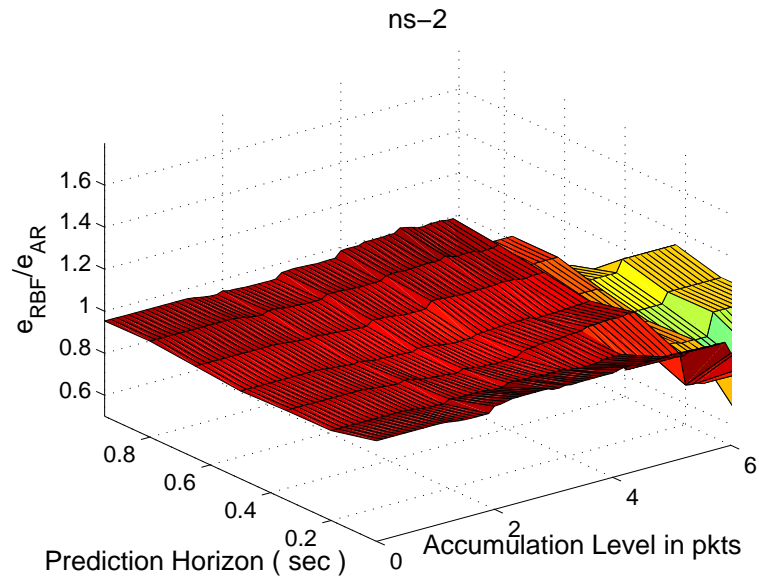


Fig. 20. Comparison of RBF and AR predictors for ns-2 simulation.

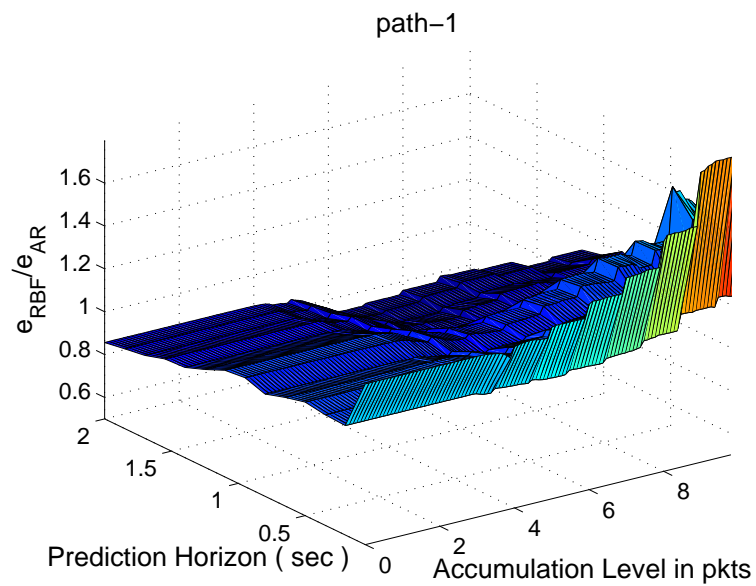


Fig. 21. Comparison of RBF and AR predictors on path-1.

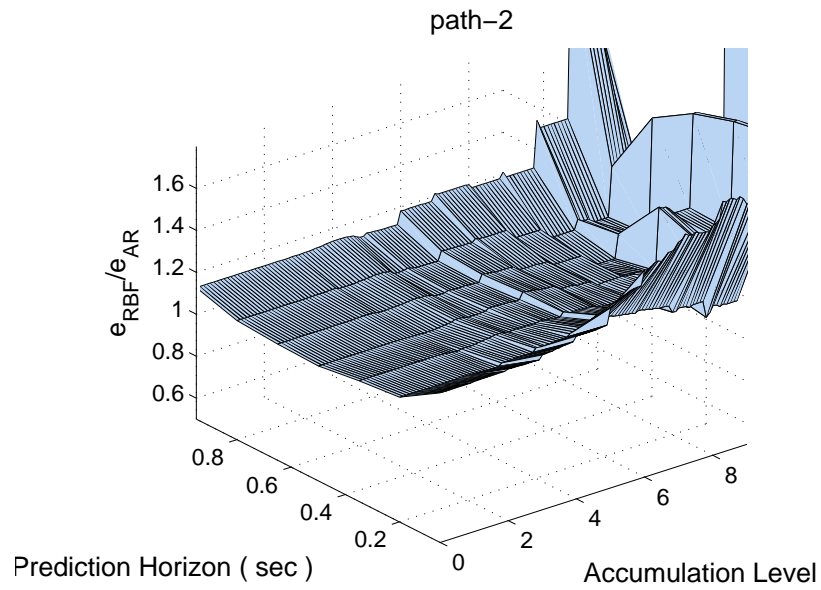


Fig. 22. Comparison of RBF and AR (order 350) predictors on path-2.

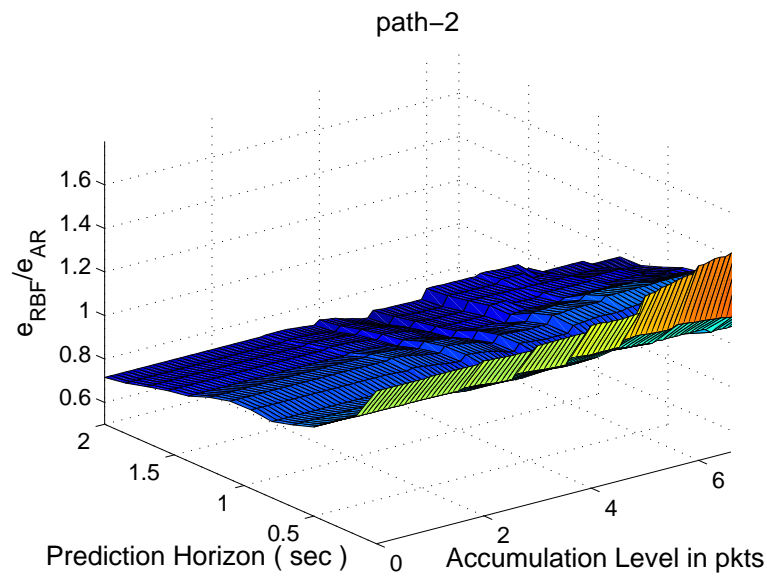


Fig. 23. Comparison of RBF1 and AR (order 40) predictors on path-2.

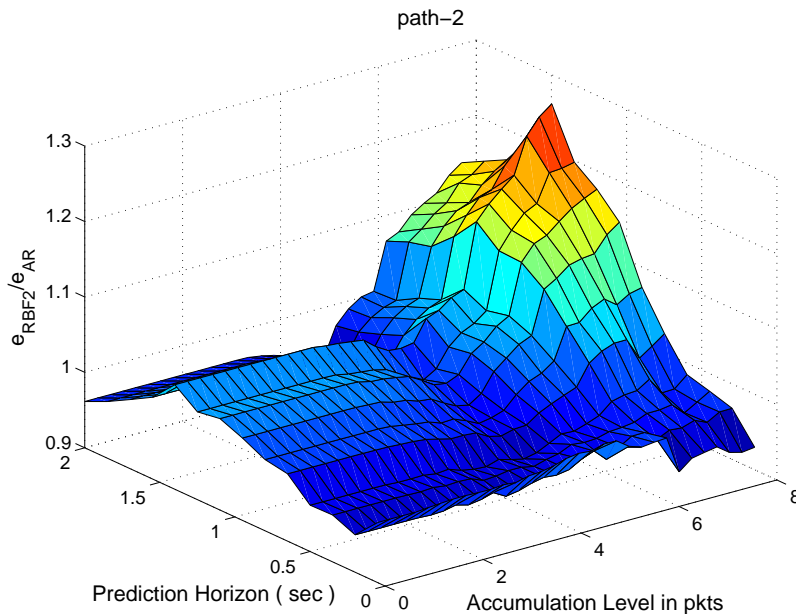


Fig. 24. Comparison of RBF2 and AR (order 40) predictors on path-2.

## G. Sparse Basis Functions

This section first deals with the key parameters involved in the design of predictors based on Sparse Basis Functions (SBF) and then presents the prediction accuracy of these models as compared with that of a Simple Predictor.

### 1. Parameter Selection

Figure 30 shows the variation of the Noise-to-Signal ratio of SBF predictors when the number of initialization samples and the forgetting factor are varied. The key points to be observed from the Figure 30 are as follows. The key parameter is the forgetting factor. The *NSR* drops almost linearly with increasing forgetting factor. This implies that SBF needs more memory to capture the dynamics of the network. Since the *NSR* drops linearly, a point close to 1 needs to be chosen. The number of

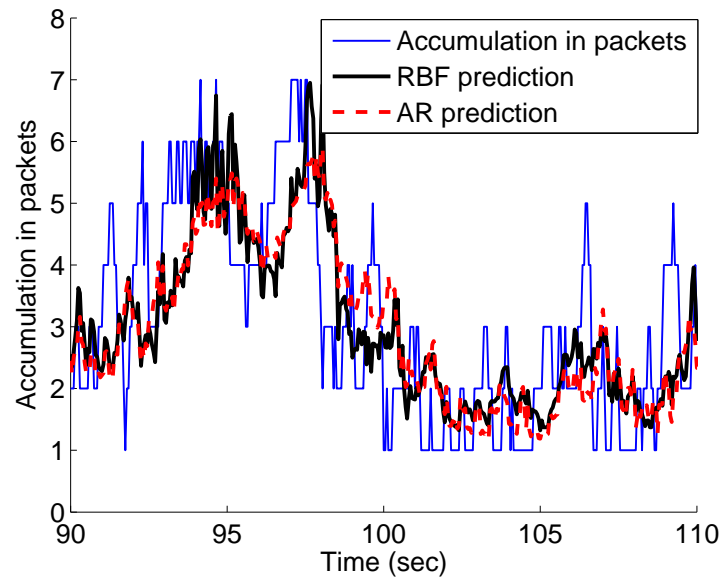


Fig. 25. 0.5 second prediction using RBF and AR predictors for ns-2 simulation scenario.

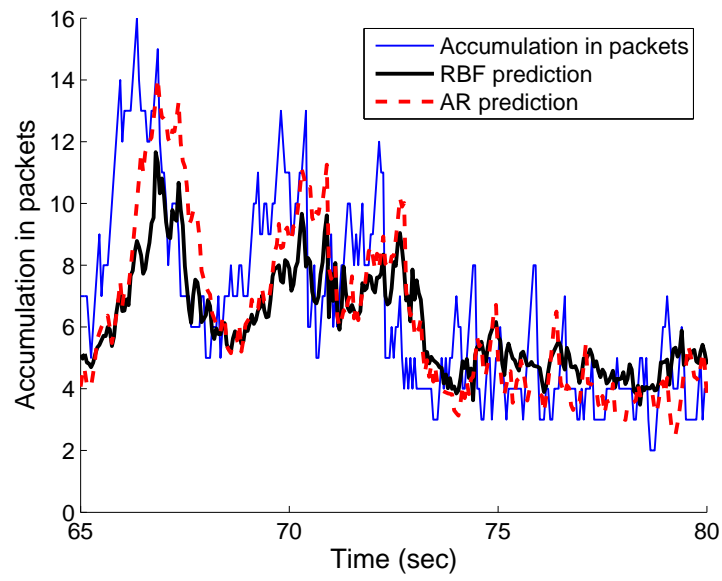


Fig. 26. 0.5 second prediction using RBF and AR predictors on path-1.

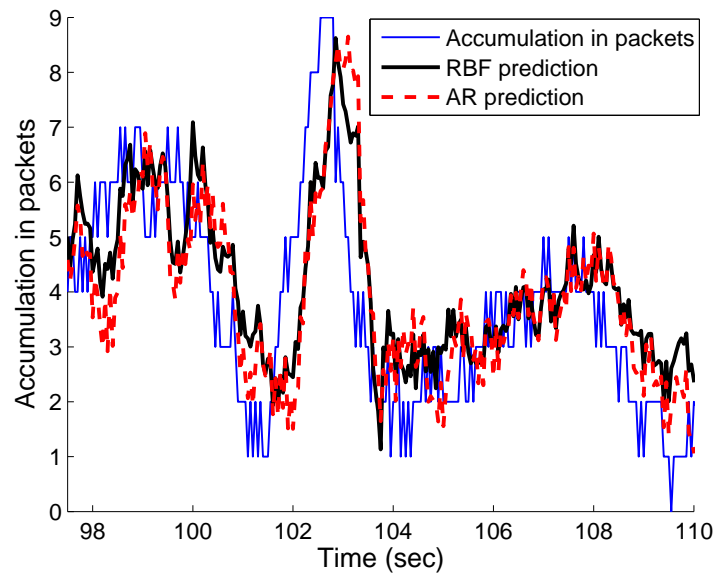


Fig. 27. 0.5 second prediction using RBF and AR predictors on path-2.

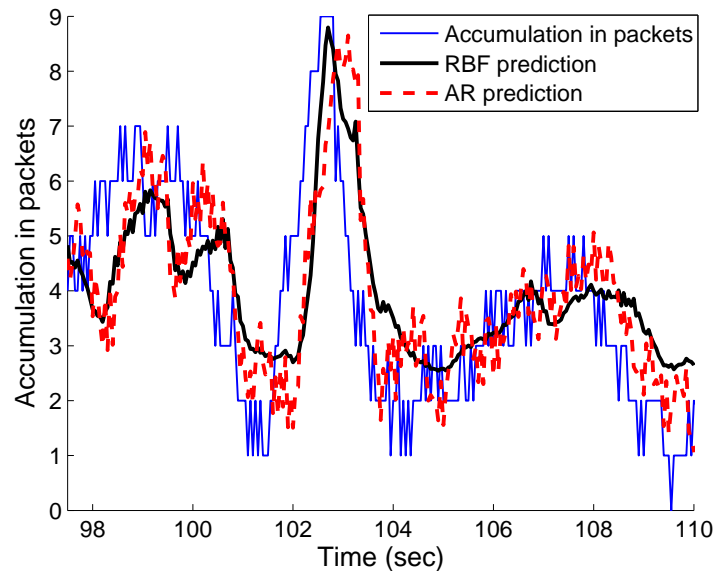


Fig. 28. 0.5 second prediction using RBF2 and AR predictors on path-2.

initialization samples has a secondary effect but nevertheless, if the forgetting factor is close to 1. Increasing the initialization samples from 200 to 400 drops the *NSR* by 50%, and beyond this point it remains constant. This can be partly attributed to the fact that the autocorrelation plot for path-2 shows a peak at 300 lags. Therefore, if the initialization points include any 300 continuous samples, the SBF performance improves drastically. The same trend can not be observed if the forgetting factor is between 0.9 and 0.96. This is because even though we include more than 300 samples, their effect on the choice of the basis drops exponentially. For example, when using a forgetting factor of 0.9, the error due to a point at 300 lags is  $0.9^{300} = 1.810^{-14}$ , when compared to a latest point being given a weight equal to 1.



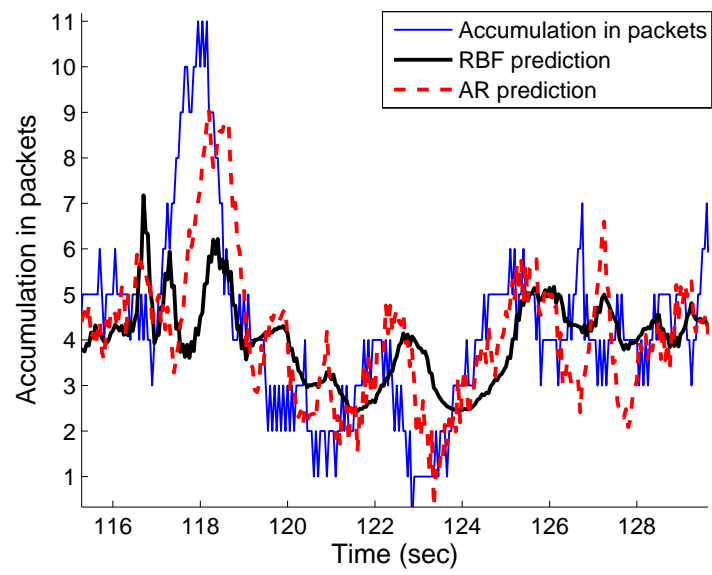


Fig. 29. 0.5 second prediction using RBF2 and AR predictors on path-2, another segment of the signal.

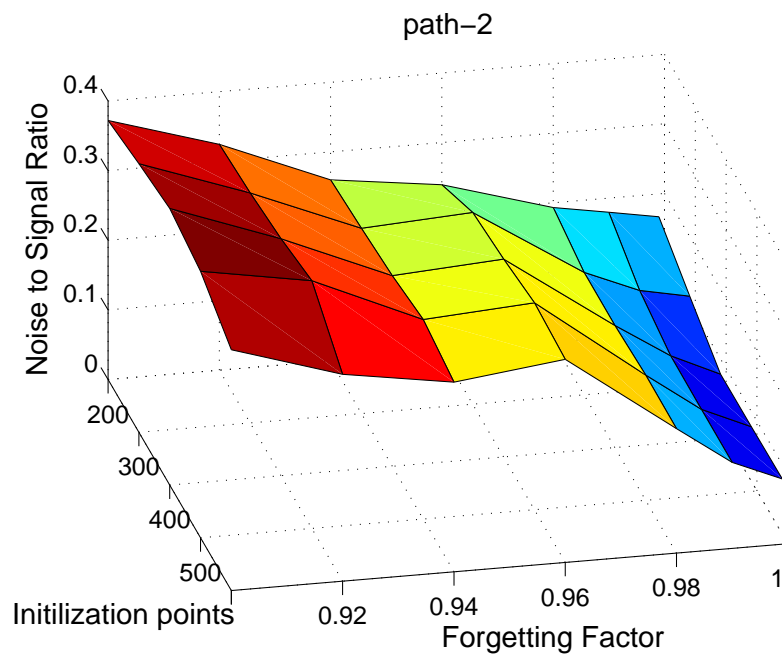


Fig. 30. Effect of the number of initialization points and the forgetting factor on the predictor  $NSR$  using sparse basis functions.

## 2. Sparse Basis Function Prediction Results

Figures 13, 14 and 15 show the prediction accuracy of the AR, RBF and SBF prediction. The AR and RBF prediction performance has already been discussed. In the case of ns-2 simulated prediction, the SBF performance was comparable to SP. Beyond 0.8 second prediction horizon, the prediction was even worse than SP. In the case of path-1, SBF shows some promise. Beyond 1 second prediction horizon, its prediction was more accurate than any of the predictors. In the case of path-2, SBF prediction was worse than AR and RBF predictors but better than SP.

Figures 31 and 32 show the prediction accuracy of Sparse Basis Functions (SBF) when compared with Simple Predictors (SP) with increasing accumulation levels. On these two paths it can be noticed that SBF performs much worse than a SP resulting in about 200% relative error. Figures 33 and 34 show the comparison of SBF with SP and AR models on path-2 respectively. In this case, the SBF performance is comparative to that of SP but when compared with an AR model, SBF results in 150% error.

Figures 35, 36 and 37 show the 0.5 second prediction of the accumulation signal for ns-2 simulation and path-1 and path-2. In all the three cases, the prediction of either just comparable to the Simple Predictor or worse. Moreover, a visual comparison with Figures 25, 26 and 27 reveals that SBF prediction is not as smooth as the AR or RBF predictions.

## H. Chapter Summary

This chapter presented an overview of the experimental set-up used in this research and the results of time-series prediction of the accumulation signal. The experiments were conducted on an ns-2 simulator and on two paths across the Internet. The real-

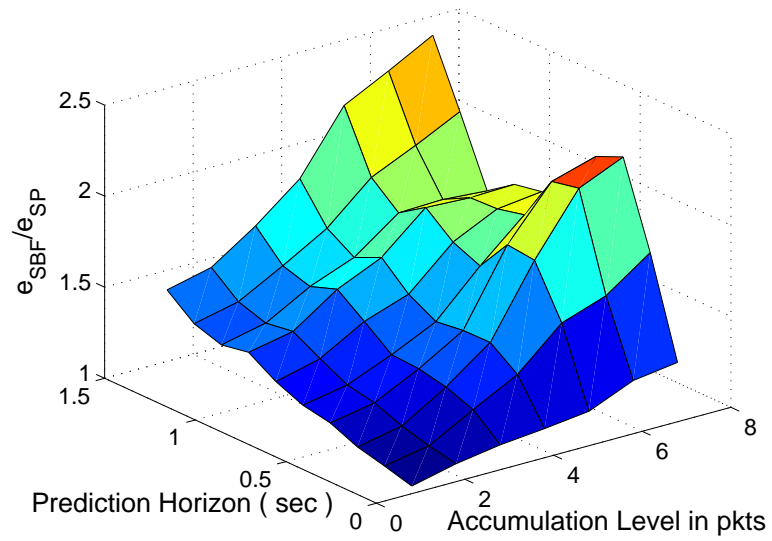


Fig. 31. SBF vs SP on ns-2 simulated data.

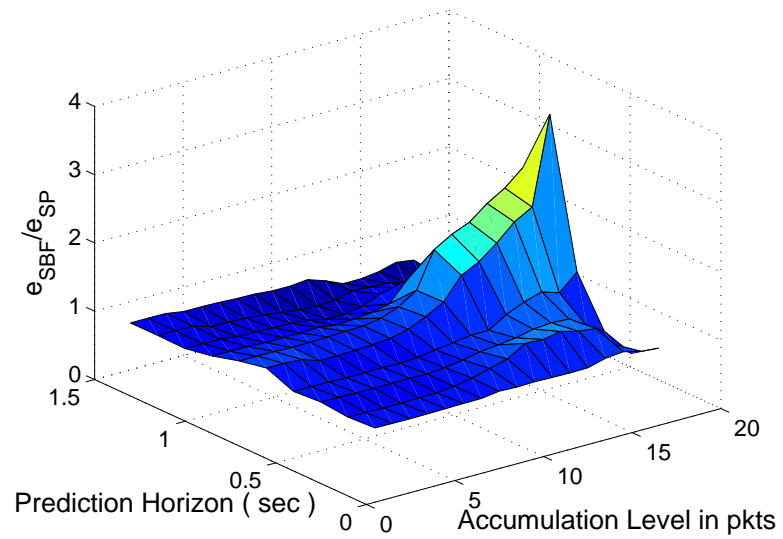


Fig. 32. SBF vs SP on path-1.

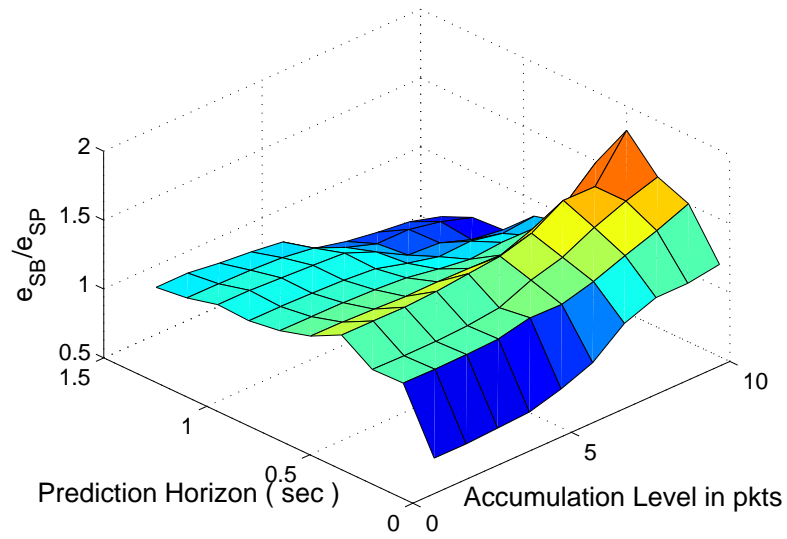


Fig. 33. SBF vs SP on path-2.

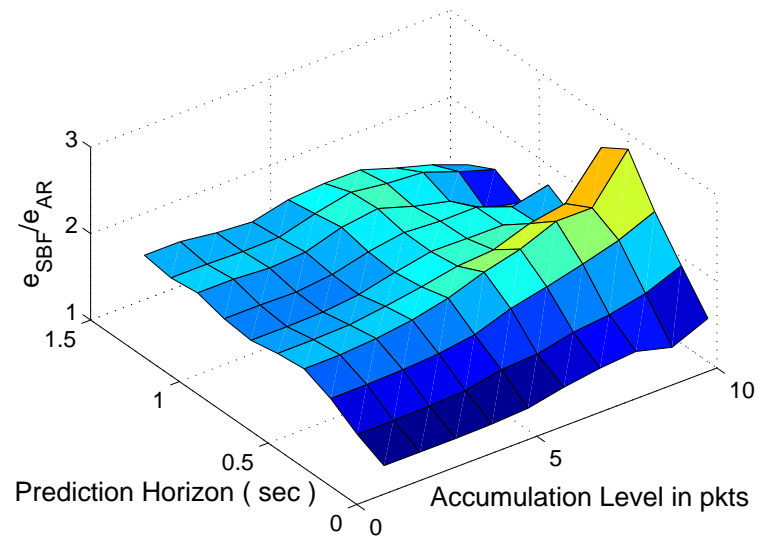


Fig. 34. SBF vs AR predictor on path-2.

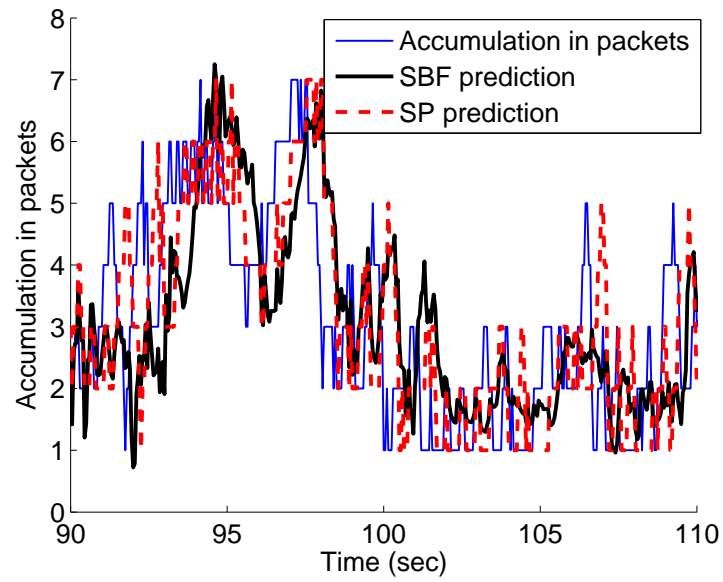


Fig. 35. 0.5 second prediction using SBF and SP Predictors for ns-2 simulation scenario.

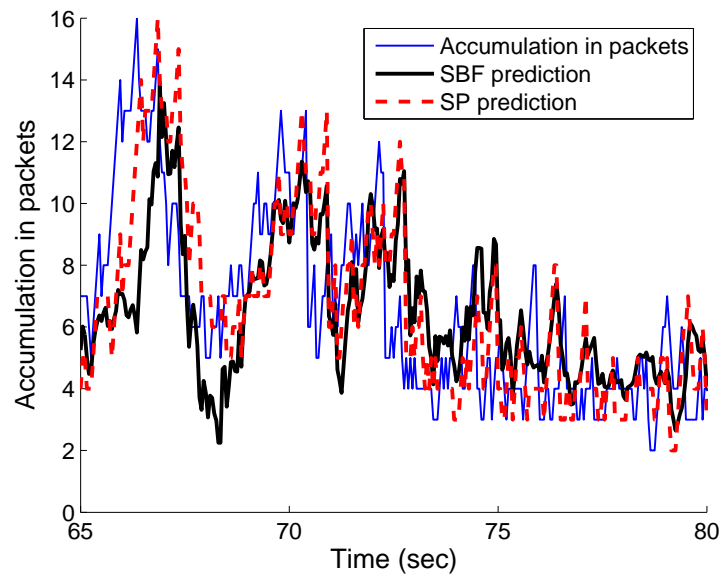


Fig. 36. 0.5 second prediction using SBF and SP Predictors on path-1.

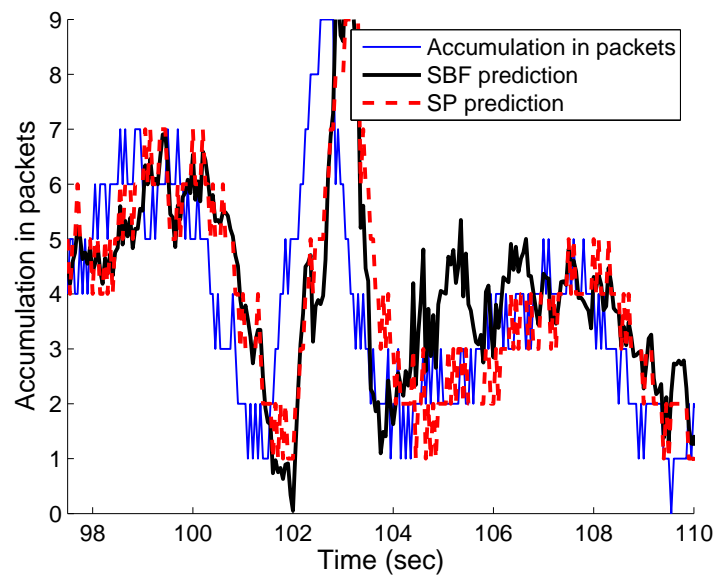


Fig. 37. 0.5 second prediction using SBF and SP Predictors on path-2.

world experiments (path-1 and path-2) were chosen to run on DSL lines to capture the behavior observed by most Internet users. The ns-2 simulation scenarios were designed to mimic the kind of traffic composition and end-to-end delay on the real-world network experiments. But still, the auto-correlation plots of the accumulation signal revealed that the ns-2 simulations do not exactly represent the dynamics of the real-world network experiments. On ns-2 simulations, the autocorrelation drops much more slowly. It also stays close to zero beyond 200 lags (10 seconds). In the case of path-1 and path-2 experiments, the autocorrelation is close to 0.1 even after 10 or 15 seconds, even though it drops to 0.2 within 50 lags (2 seconds).

Time-series models of the accumulation signal were developed using AutoRegressive models (AR), Radial Basis Functions (RBF) and Sparse Basis Functions (SBF). The prediction of these models was then compared with Simple Predictor (SP) as the baseline. It was found that fine tuning RBF models results in the least *NSR* of all the models. But *NSR* does not reflect the prediction accuracy of the predictors at high accumulation levels. Higher accumulation levels represent congestion and are the most important durations when accurate prediction is most important. Therefore, a second method of comparison based on accuracy with increasing accumulation levels revealed that RBF predictor is still the best for ns-2 and path-1 cases. For accumulation prediction on path-2, AR model of order 350 (17.5 seconds) was found to be the best predictor. But high order implies higher computational time required for developing the model. An RBF model with 10 lags (0.5 seconds) was found to give a prediction accuracy quite comparable to the AR model.



## CHAPTER IV

## EFFECT OF BIT-RATE ON ACCUMULATION SIGNAL

## A. Introduction

To improve end-to-end QoS of applications deployed over the Internet, the control algorithm must be able to effect the system through a control input. Source bit-rate is the most logical and intuitive input that may be chosen for a wide range of applications. Source bit-rate is analogous to process control inputs such as mass flow of a chemical into the process, in the case of a chemical processing plant. In these systems, once the inputs enter the system, there is no way to retrieve them back. This is the key feature that makes predictive control schemes very effective in such systems. Therefore, this chapter analyzes the sensitivity of the network to source bit-rate variations. A very good sensitivity will imply that the system is controllable through the selected input. In real-time applications, a packet that arrives after its deadline is treated as a lost packet. Therefore, in the analysis presented below, packets having delay greater than their deadline are treated as lost packets. In all of the foregoing discussion, the term ‘loss’ reflected dropped packets and the term ‘total loss’ means the sum of dropped packets and packets which do not meet the deadline.

To analyze the effects of changes in the bit-rate, experiments were conducted where the bit-rate is switched every “T” seconds between a low value and a high value. The change in the bit-rate is brought about by changing the packet size. Figure 38 shows the way the network is probed. The bit-rate is switched between R1 and R2 bps every T seconds. The lower bit-rate R1 is chosen to have minimal effect on the network. It therefore captures the characteristics of cross traffic. The higher bit-rate R2 will reveal the effect of increasing the bit-rate on end-to-end behavior.

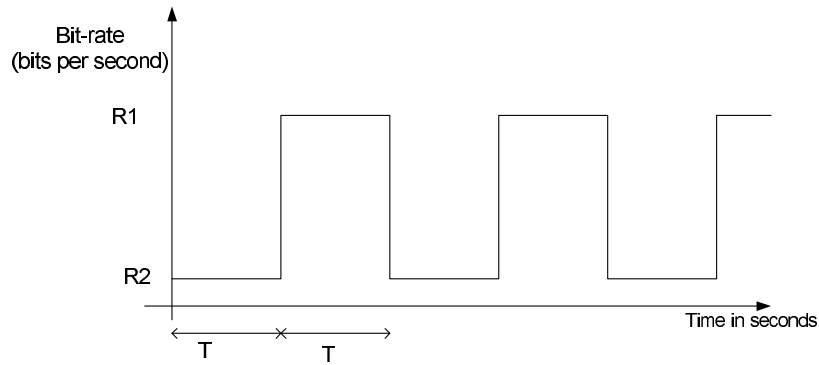


Fig. 38. Adjusted bit-rate vs time for experiments to explore the impact of bit-rate on packet loss and delayed packets.

In this chapter we present results for the above described experiment over two different paths called ‘path-1’ and ‘path-3’ (to be consistent with the path-1 and path-2 used in the previous chapter). On path-1, the end-to-end delay is high compared to the sampling interval and on path-3 the end-to-end delay is low compared to the sampling interval. The details of the experiments are given in Table III. For experiments on ‘path-1’, the source IP was planetlab1.gti-dsl.nodes.planet-lab.org and the destination IP was planetlab1.iiitb.ac.in. For this path, the bit-rate was switched between 13.12 kbps to 45.12 kbps (almost 4 fold increase). The change in bit-rate was effected by changing the packet size but keeping the inter-departure time constant at 52 milliseconds. The bit-rate was switched every 4 seconds ( $T$  seconds). For experiment on ‘path-3’, planetlab1.nbgisp.com was chosen as the source and planetlab1.gti-dsl.nodes.planet-lab.org as the destination. The bit-rate was switched between 15.52 kbps and 128.32 kbps (almost 8 fold increase) by increasing the packet size and keeping the inter-departure time fixed at 22 milliseconds. The bit-rate is switched every 22 seconds ( $T$  seconds). The choice of the switching interval and the difference in the bit-rate switching factor between the two paths are justified in the later sections.

Figures 39 and 40 show the one-way delay with sequence number of the packets in each case. The first path is characterized by high end-to-end delays. We choose a deadline of 500 ms for this path. A deadline of 500 ms is suited for applications such as networked robotics. For the second path, we set the deadline as 200 ms, usually used for VoIP and video-conferencing applications. The collected data are first analyzed for their statistical differences in the packet loss characteristics. The data are then trained with different linear and non-linear models, to investigate which models explain the statistical results and which models predict the data more accurately.

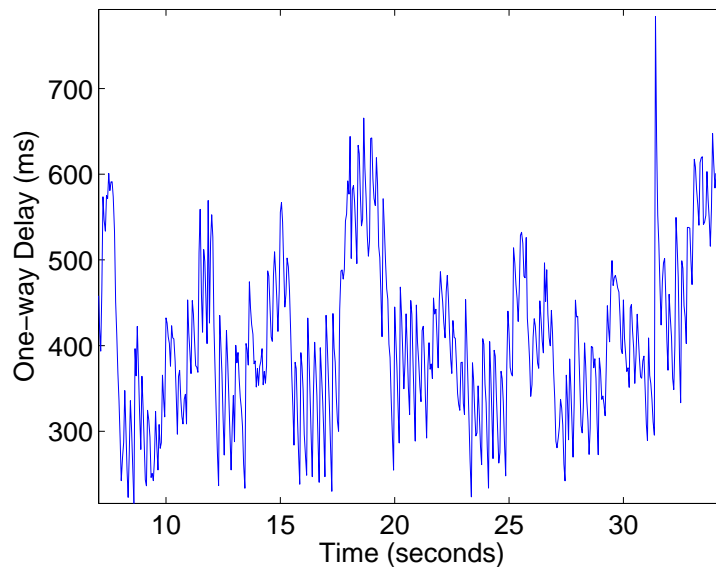


Fig. 39. One-way delay traces for path-1.

## B. Statistical Analysis

In both of the experiments, each level of bit-rate persists for  $T$  seconds before switching to the second level. The analysis on the data sets is presented using two methods.

Table III. Description of Variable Bit-Rate Experiments

<b>Name</b>	<b>Path-1</b>	<b>Path-3</b>
<b>Source</b>	planetlab1.gti-dsl .nodes.planet-lab.org	planetlab1.nbgisp.com
<b>Destination</b>	planetlab1.iiit.ac.in	planetlab1.gti-dsl .nodes.planet-lab.org
<b>R1 packet size</b> (bytes)	82	62
<b>R1 packet inter-departure time</b> (ms)	52	22
<b>R1 bit-rate</b> (kbps)	13.12	45.12
<b>R2 packet size</b> (bytes)	282	350
<b>R2 packet inter-departure time</b> (ms)	52	22
<b>R2 bit-rate</b> (kbps)	45.12	128.32
<b>Sampling time</b> (ms)	50	50
<b>Time Period of bit-rate</b> (T sec)	4	22

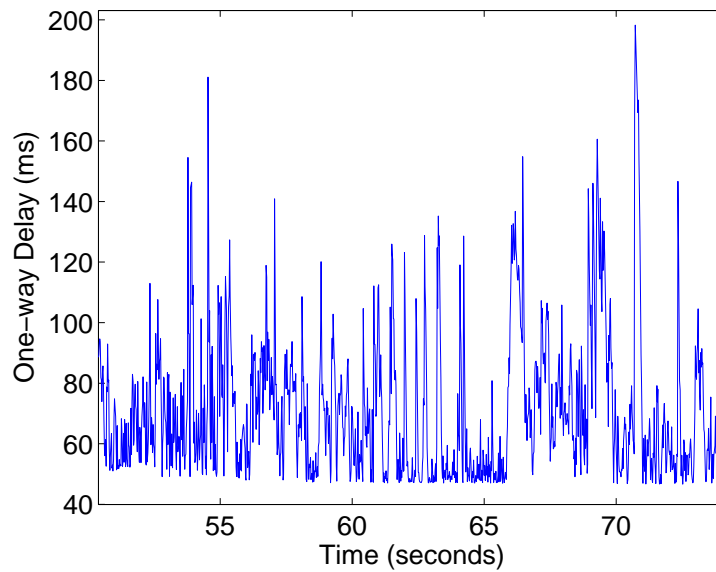


Fig. 40. One-way delay traces for path-3.

First method is to treat every  $T$  seconds of data of each bit-rate as one sample and characterize the relative difference in losses between the lower and higher bit-rates. In the second method, time-ensemble averages of the packet delays and losses are performed on  $2T$  seconds ( $T$  seconds of low bit-rate and  $T$  seconds of high bit-rate). Each  $2T$  segment of the data is collected and the average delay and packet loss across each of these sets is calculated. This would present a statistically averaged effect of increase in the bit-rate on end-to-end delay and packet loss. The primary reason for performing time-ensemble average is to analyze the effects of a change in input bit-rate on the end-to-end delay and loss dynamics.

## 1. Experiments on Path-1

Figures 41 and 42 show the cumulative distribution of packet loss and total loss (includes losses due to packets with delay greater than deadline). Suppose that the application required less than 5 % packet loss. Then from Figure 41 we see that 70% of the samples with the lower bit-rate (Bitrate-1) satisfy this criterion where as only 50% of the higher bit-rate (Bitrate-2) satisfy this criterion. For most real-time applications a packet with delay greater than the deadline is as good as a lost packet. Therefore, if the criteria is less than 5% total loss, about 55% of samples of lower bit-rate satisfy the requirement where as only about 20% of samples of higher bit-rate (Bitrate-2) meet the specification. This clearly shows that lowering the bit-rate can have drastic impact on real-time application behavior. Figure 43 shows the scatter plot for the total losses when using the lower and higher bit-rates. Each point on the plot is obtained by finding the losses in a  $T$  second interval corresponding to lower bit-rate and then computing the losses in the next  $T$  second interval (higher bit-rate samples). From Figure 43 it is evident that for any given total loss for the lower bit-rate, the total loss for the higher bit-rate is more likely to be higher. For some samples, the total loss of the lower bit-rate is high, but when seen from a statistical point-of-view, this is more an aberration than a rule. Such outliers are due to different cross-traffic during each of the  $T$  second intervals.

To analyze the dynamic effects of change in the bit-rate, we perform time-ensemble average of the one-way delay and total losses. The results are presented in Figures 44 and 45. It can be noticed from the delay plot that when the bit-rate is changed from low to high, the delay increases and it takes about 1 second to reach close to the steady-state value. This justifies our choice of 2 seconds as the switching time. The results also suggest that if the controller switches faster than 1 sec,

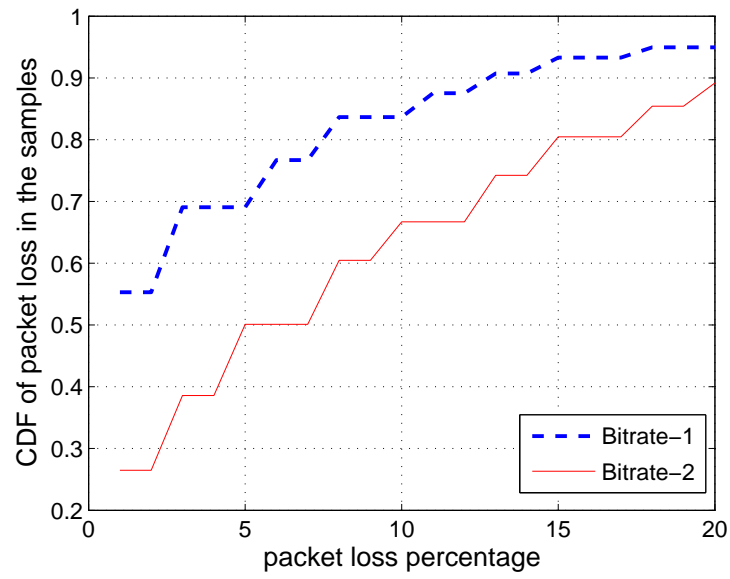


Fig. 41. CDF for packet loss in the experiment on path-1.

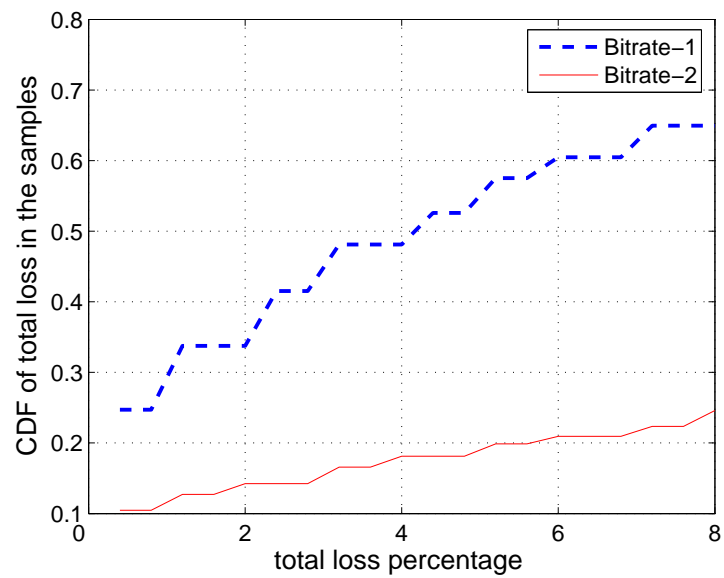


Fig. 42. CDF for total loss in the experiment on path-1.

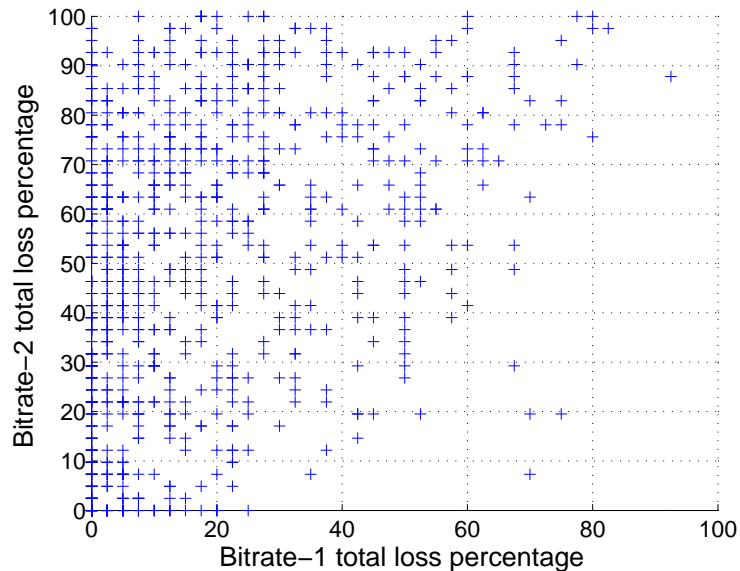


Fig. 43. Scatter plot for total loss when using two bit-rates in the experiment on path-1.

the effect of bit-rate change may not be evident. The time-constants involved in the system are clearly evident from Figure 45. From this figure we find that the settling time for system is about 1 sec. The figure also shows that the total loss can decrease from 50% to about 15% by changing the bit-rate.

## 2. Experiments on Path-3

Figure 46 and 47 show the cumulative distribution of packet loss and total loss for path-3. On the path used for this experiment, we find very low losses as is evident from Figure 46. The higher bit-rate actually had less packet losses than the lower bit-rate. But it should be noted that the packet loss in both the cases is quite negligible (about 0.1 %). The effect of bit-rate is clearly evident when we consider total losses. As in the analysis for path-1, if we consider 5% as the acceptable total loss, 90% of lower bit-rate samples satisfy the criteria compared to 83% for the higher bit-rate



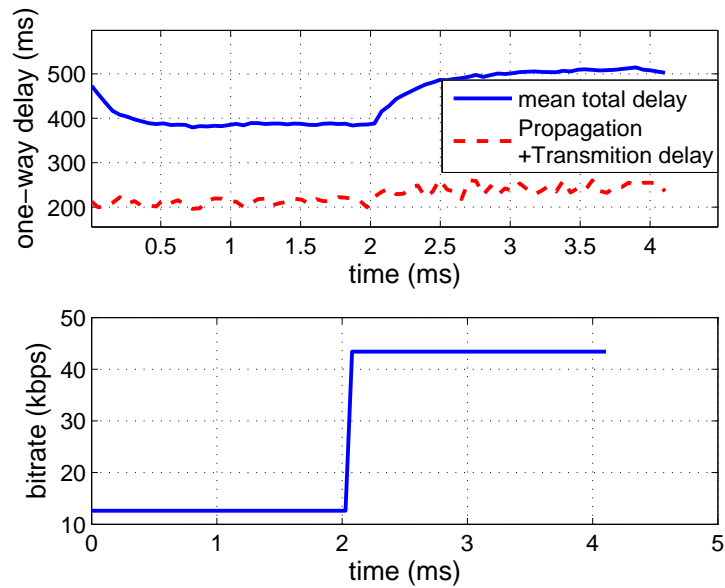


Fig. 44. Time-ensemble average of the one-way delay on path-1.

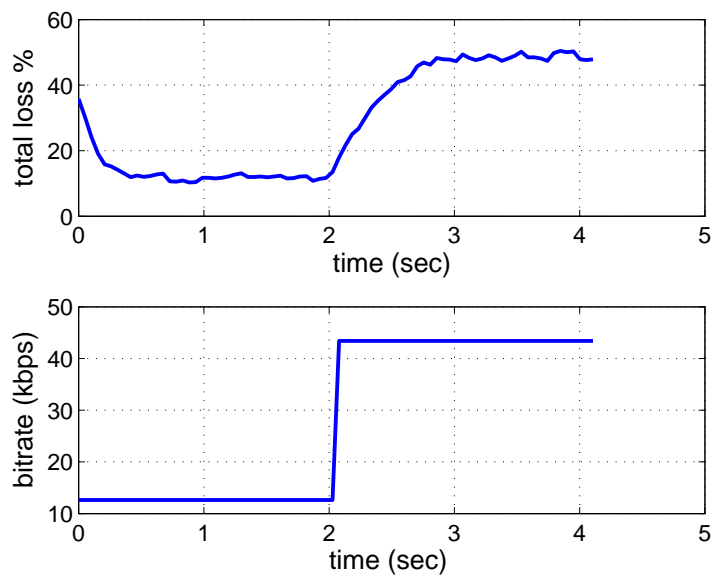


Fig. 45. Time-ensemble average of the total loss on path-1.

case. The change in the bit-rate is of about 5 times, same as in the first experiment, but only that both the bit-rates are relatively higher. The scatter plot shown in Figure 48 shows that that the higher bit-rate has higher losses compared to the lower bit-rate. But the samples are more scattered when compared with the scatter plot for the path-1. One of the main reasons for this is that the duration  $T$  of each bit-rate is about 25 sec compared to 2 sec used in path-1. Therefore, the samples for the two bit-rates tend to experience different network congestion. The reason for choosing such a high time-period is evident from the time-ensemble results. Figures 49 and 50 show the time-ensemble average delay and loss for one time-period duration. From these figures, it is evident that for this path the settling time is close to 5 sec. Moreover, the system exhibits a more complex behavior compared to the first path. The first path could have been approximated using a first order system but in this case, the jump at 3 seconds on the Figures 49 and 50 indicated a higher order system.

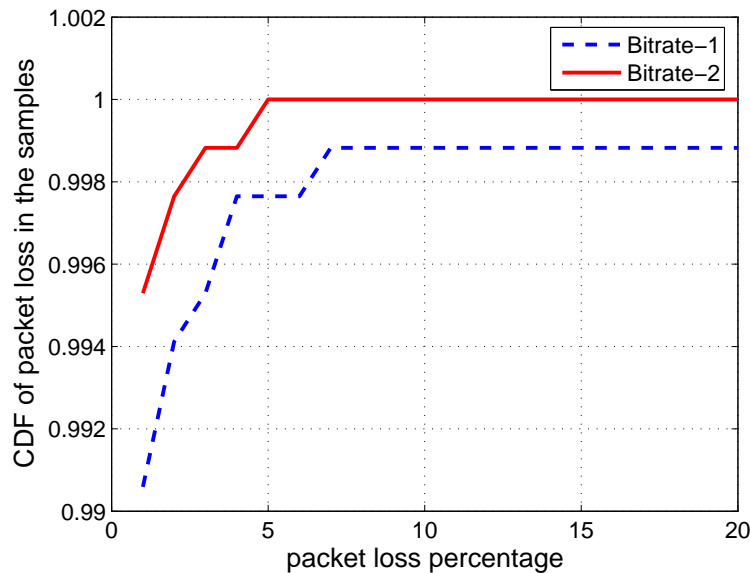


Fig. 46. CDF for packet loss in the experiment on path-3.

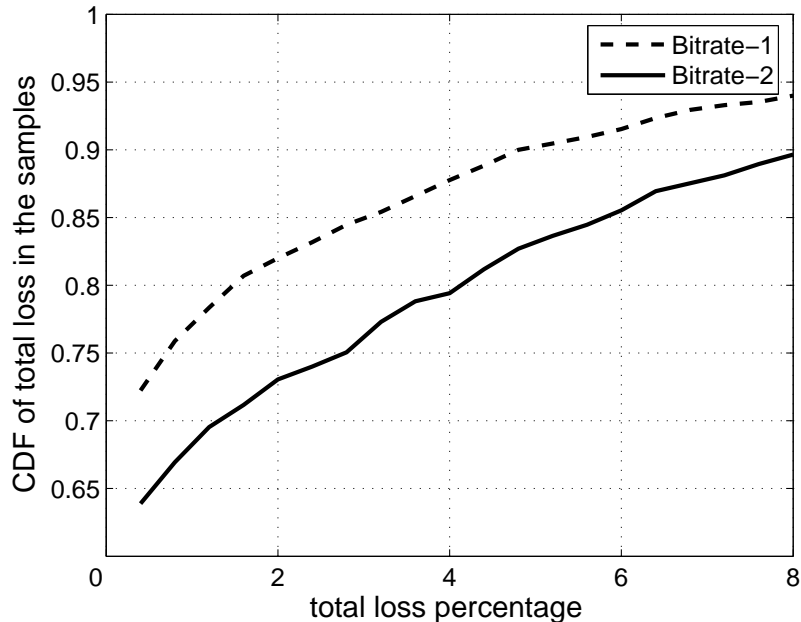


Fig. 47. CDF for total loss in the experiment on path-3.

### C. Prediction of Accumulation Signal

The accumulation signal was defined to be the number of bytes accumulated in the network. In the case of constant bit-rate, with constant packet size, accumulation in packets is equivalent to accumulation in bytes because the two differ only by a factor equal to the packet size. In this section, we present various models that are used to predict the accumulation in bytes. In addition, we also investigate the predictability of accumulation in packets due to the following reason. In the previous section, analysis was carried out to ascertain the statistical effects of change in bit-rate on ‘packet loss’ and ‘total loss’. During this process, the loss and one-way delay of each packet was treated as a single sample, irrespective of its size in bytes. Therefore, if one predictor is able to predict the accumulation in packets more accurately than an

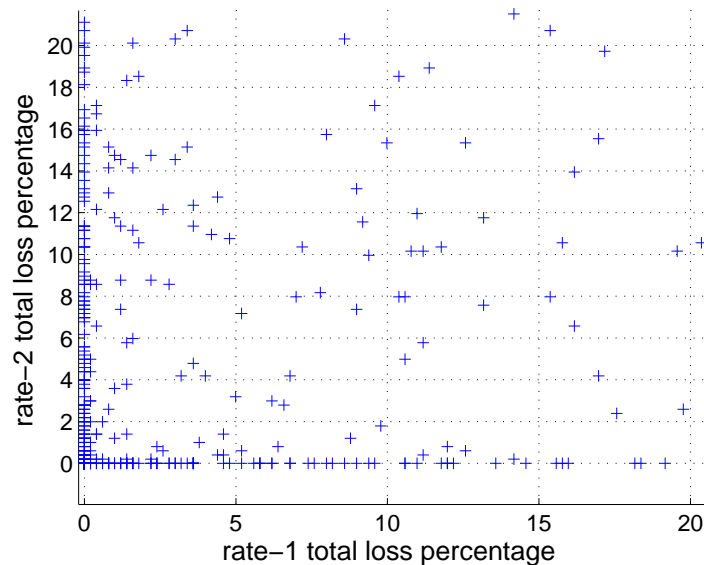


Fig. 48. Scatter plot for total loss in the experiment on path-3.

another predictor, we can infer that it is also able to explain the statistical differences observed in the previous section better than the other predictor.

### 1. Parameter Selection

The Akaike Information Criterion (AIC) is used in selecting the parameters of the AR model order. In obtaining the AIC for each model, we train with 5000 samples and select the best model order. For an ARMAX, the parameters are  $na, nb, nc, nk$ , each being the order of the polynomials A,B,C shown in 2 and the delay in the feedback loop respectively. The AR ( $na$ ), ARMA( $na, nc$ ), ARX( $na, nb, nk$ ) models can be seen as specializations of the ARMAX model. For each experimental case, the whole domain needs to be searched for the model order combination that minimizes the AIC. The search can be reduced by noting that the experiments were designed by

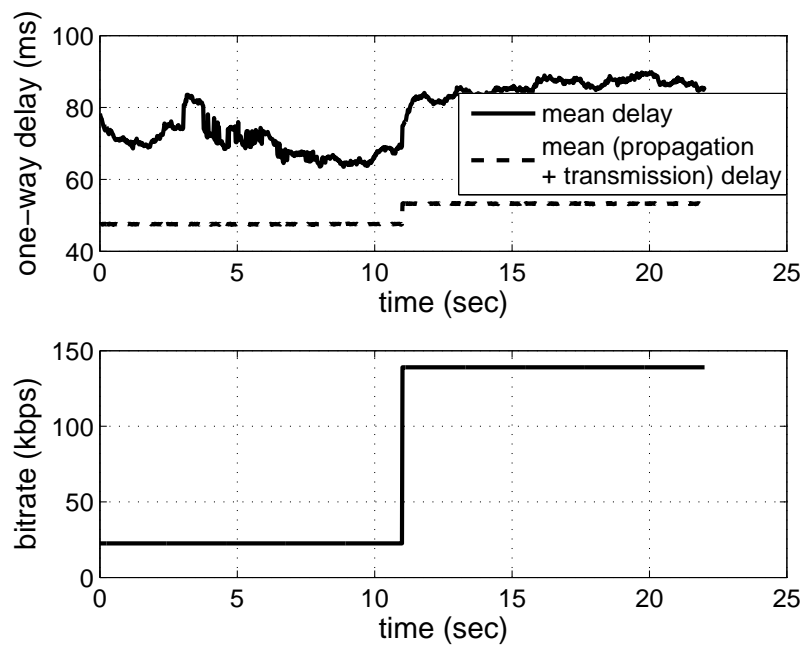


Fig. 49. Time-ensemble average of the one-way delay on path-3.

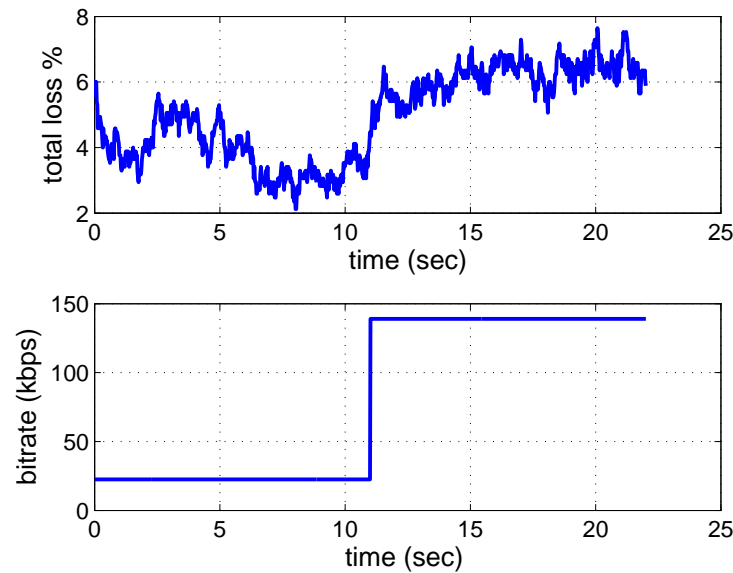


Fig. 50. Time-ensemble average of the total loss on path-3.

assuming that the effects of a bit-rate change are negligible after half the time period of the bit-rate change. Therefore, if the bit-rate time period is  $N$  samples, the search can be limited to  $N/2$  for any of the orders.

The input vector for the RBF models is  $[AB]^T$  where  $[A] = [a(k)a(k-1)..a(k-p)]$  represents the accumulation levels,  $[B] = [b(k)b(k-1)..b(k-p)]$  represents the bit-rate and 'p' is the order of the RBF model. For an 'n' step prediction case, the target to be matched is  $a(k+n)$ . The RBF model parameters were determined by the procedure described in the previous chapter on time-series modeling. Figures 51 and 52 show the effect of parameter variation on the accuracy of the Radial Basis Function (RBF) predictor performance. It should be noted that the first approximation of the radius can be obtained by calculating the average norm-2 distance between the input vectors, which is close to 1000 when the accumulation is in bytes. Figures 53 and 54 show

the effect of parameters of RBF when predicting accumulation in packets. When the accumulation is in bytes, we can expect RBF network to perform better than a SP. When accumulation is in packets, RBF performs much better than SP in path-1 but in path-3 we have no clear advantage. Tables IV and V show the final models selected for predicting accumulation in bytes and packets, respectively.

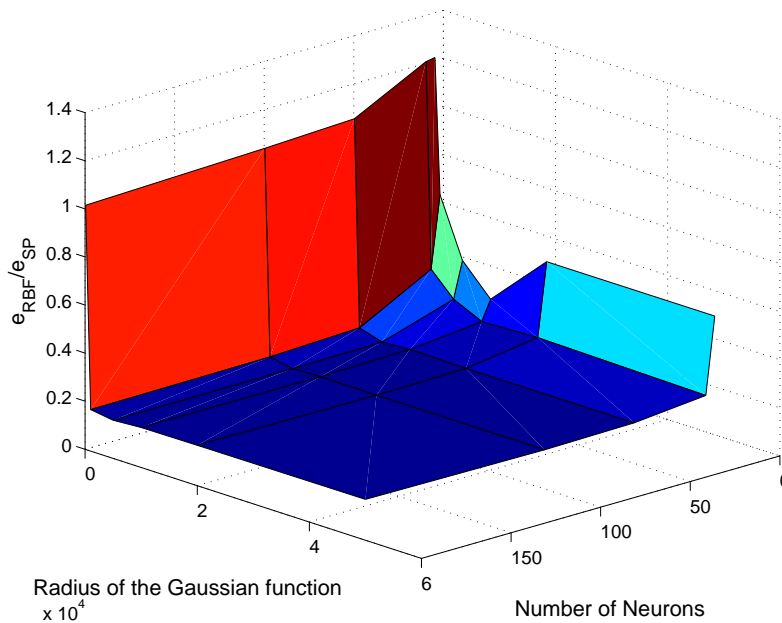


Fig. 51. Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-1 with accumulation in bytes.

## 2. Results for the Prediction of Accumulation in Bytes

### a. Experiments on Path-1

Figure 55 shows the  $NSR$  when using a SP and also shows the ratio of errors when using AR, ARMA, ARX, ARMAX and RBF predictors as compared to SP with increasing prediction horizons. It can be noticed that the  $NSR$  increases almost

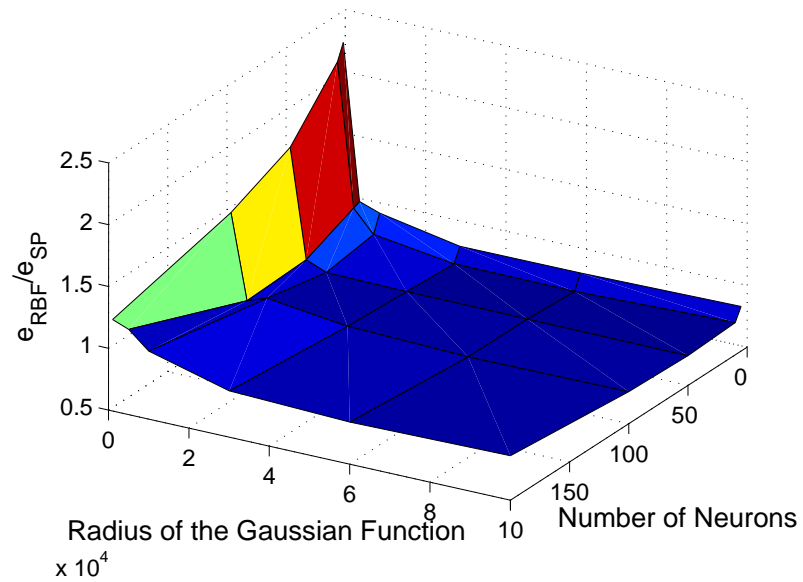


Fig. 52. Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-3 with accumulation in bytes.

Table IV. Model Parameters (Accumulation in Bytes)

Model	Path-1	Path-3
AR order	33	40
ARMA order	(31,31)	(13,13)
ARX order	(10,13,1)	(61,13,1)
ARMAX order	(31,13,28,1)	(49,49,25,1)
RBF		
Neurons	50	50
Radius	20000	40000
Order	10	10



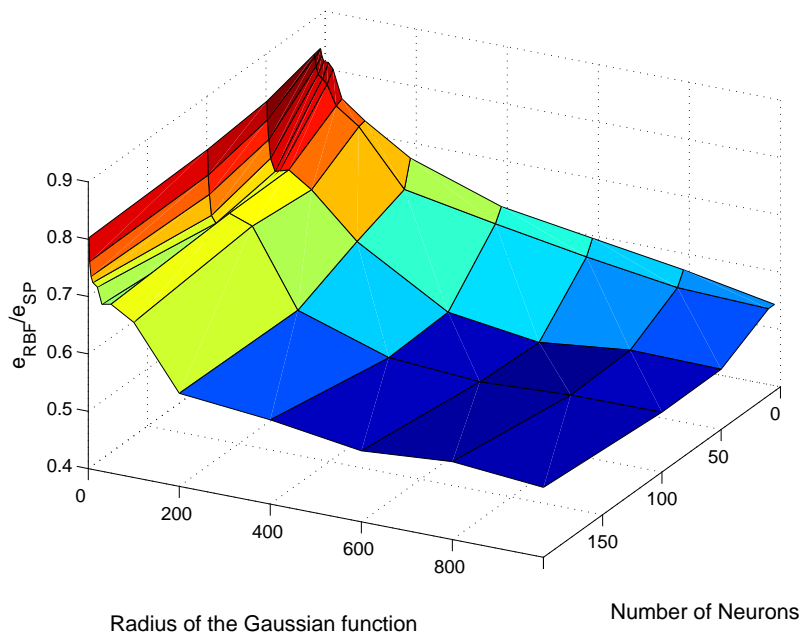


Fig. 53. Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-1 with accumulation in packets.

Table V. Model Parameters (Accumulation in Packets)

Model	Path-1	Path-3
AR order	13	40
ARMA order	(22,10)	(19,31)
ARX order	(25,10,1)	(43,13,1)
ARMAX order	(22,22,13,1)	(31,19,19,1)
RBF		
Neurons	50	50
Radius	600	1000
Order	10	10

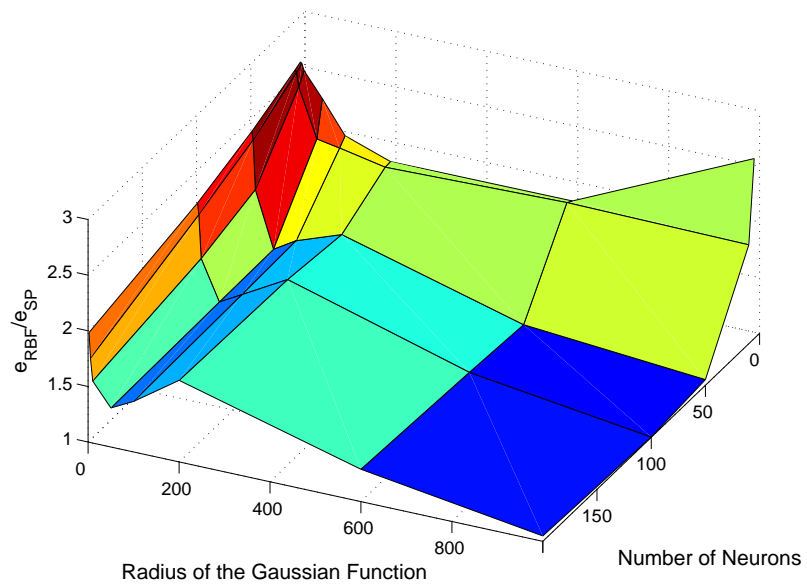


Fig. 54. Effect of the radius of the Gaussian weight and the number of neurons on prediction error with RBF network in the experiment on path-3 with accumulation in packets).

linearly with time when using SP. It can also be noted that an AR predictor does not improve much when compared to the SP. The error ratio of AR to SP is close to 0.8 with increasing prediction horizon. The interesting fact to be observed is that the ARMA model drastically improves over the AR model and at about 1 sec prediction horizon, the error ratio of ARMA to SP is close to 0.2. ARX, ARMAX and RBF predictors show a very comparable prediction accuracy with RBF consistently performing slightly better than the linear predictors ARX and ARMAX.

The predictors are then analyzed for their performance with increasing accumulation levels. Figure 56 shows the comparison of AR and SP. The plot shows that at lower accumulation levels, the AR model is slightly better than a SP. As the accumulation level increases, AR prediction is comparable to that of the SP. Figure 57 compares the ARMA model with an AR model. The figure shows that as the prediction horizon increases, ARMA model performs much better than an AR model. But at higher accumulation levels, the ARMA prediction degrades slightly. Figure 58 shows the relative performance of ARX model when compared to an AR model. It can be noticed that the relative ARX prediction accuracy improves with increasing prediction horizon. Figure 59 shows ARMAX performance when compared to AR model. ARMAX model does not show any significant improvement over the ARX model. Figure 60 shows the relative performance of RBF network when compared to an AR predictor. It can be noticed that RBF out-performs the AR predictor. But from Figure 56 it can be noticed that the accuracy is comparable to that of ARX. Hence for this system, an ARX model is sufficient.

Figures 61, 62 and 63 show the time-series plot of the prediction using AR, ARX, ARMA, ARMAX predictors. It can be noticed that ARMA, ARX and ARMAX predictors have very comparable performance. Moreover, the effect of the input bit-rate change is clearly visible in that the models that include exogenous input (bit-

rate) closely follow the rising and falling edges of the signal. Figure shows that RBF performs better than AR predictor but as discussed in the preceding paragraph, the accuracy is quite comparable to that of ARX.

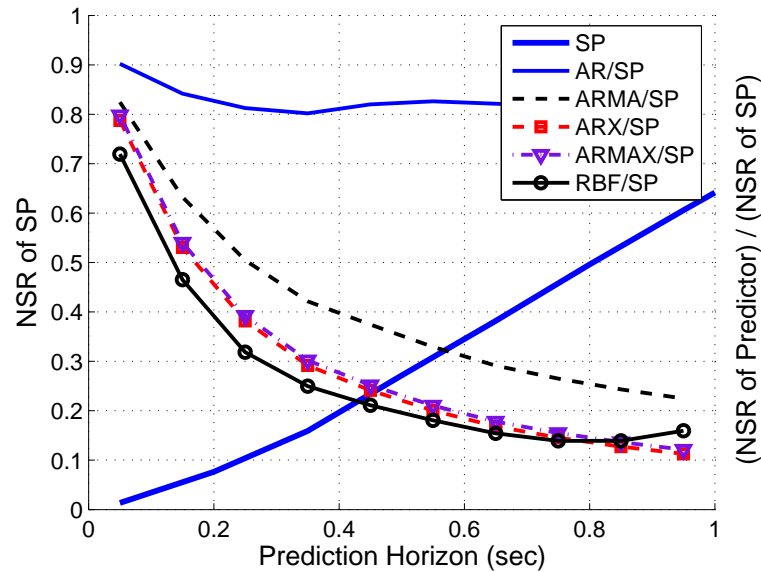


Fig. 55. Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for path-1.

#### b. Experiments on Path-3

Figure 64 shows the  $NSR$  when using a Simple Predictor and also shows the ratio of errors when using AR, ARMA, ARX, ARMAX and RBF predictors as compared to SP with increasing prediction horizons. It can be noticed that the  $NSR$  increases almost linearly upto 2 seconds prediction and then settles at 1.4 after 3.5 seconds. Since  $NSR$  of SP crosses 1 at 1.5 seconds, SP is not useful beyond this prediction horizon. The AR and ARMA predictors show a slight improvement over the SP but the improvement is only 0.7 and 0.8 respectively at 5 seconds prediction horizon. Based on the  $NSR$

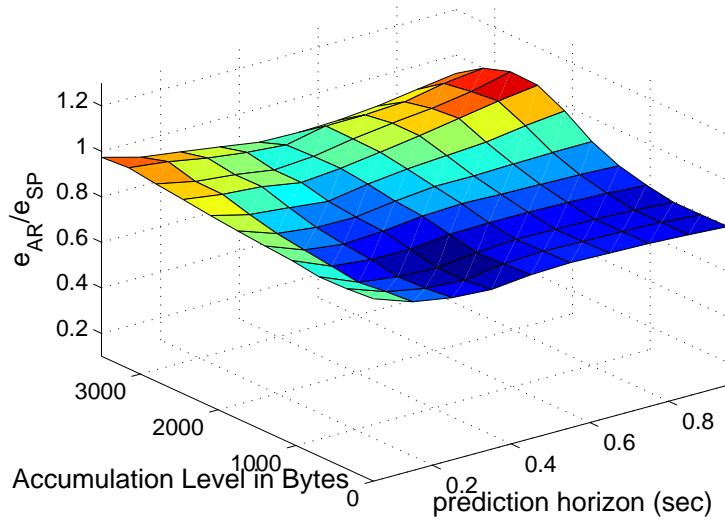


Fig. 56. Prediction of byte-accumulation with AR and SP on path-1.

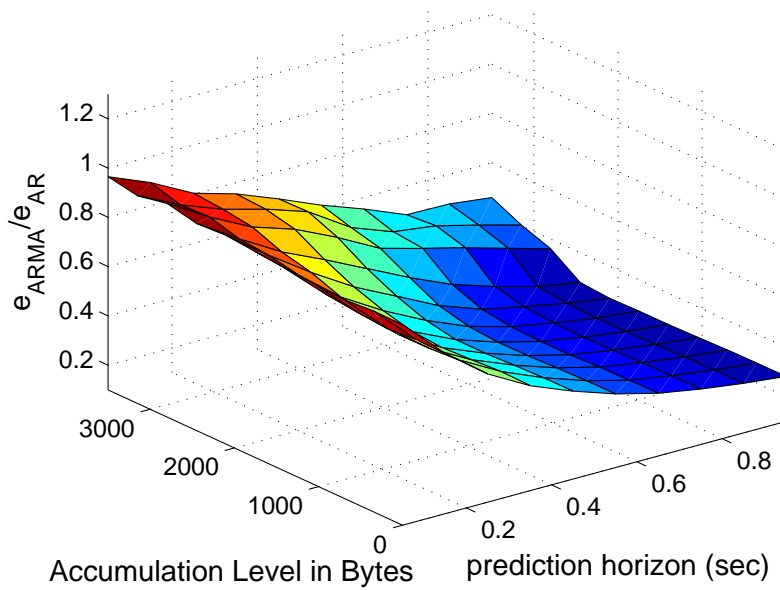


Fig. 57. Prediction of byte-accumulation with ARMA and AR on path-1.

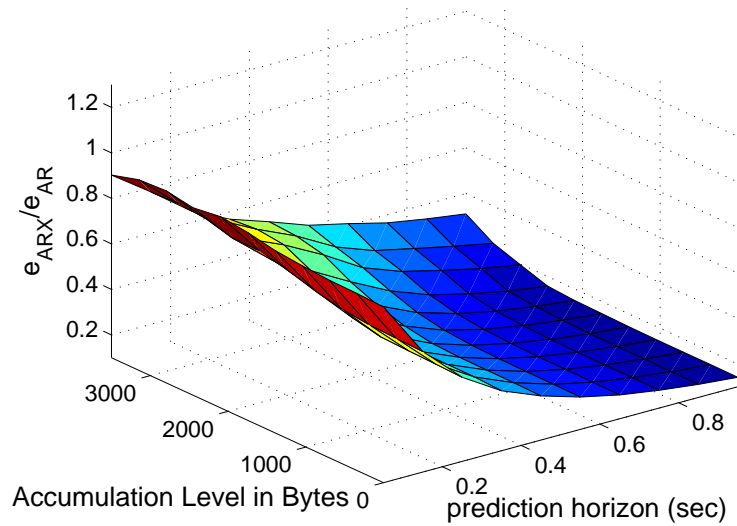


Fig. 58. Prediction of byte-accumulation with ARX and AR on path-1.

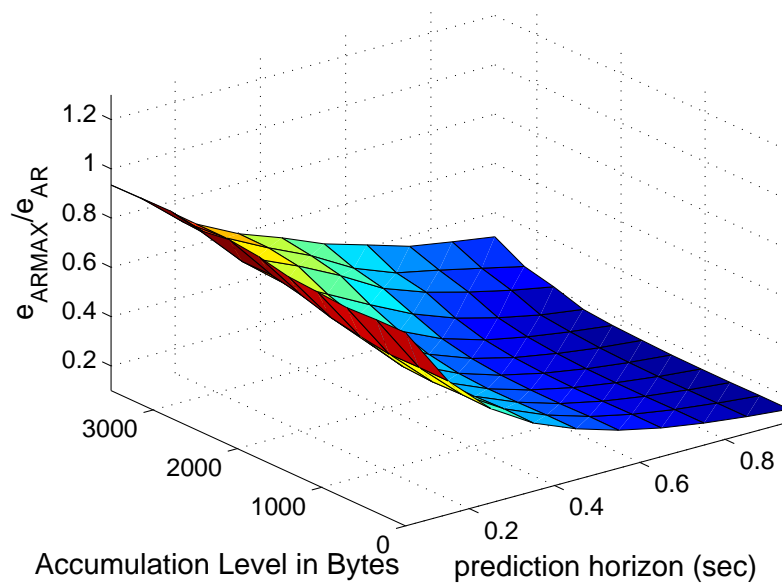


Fig. 59. Prediction of byte-accumulation with ARMAX and AR on path-1.

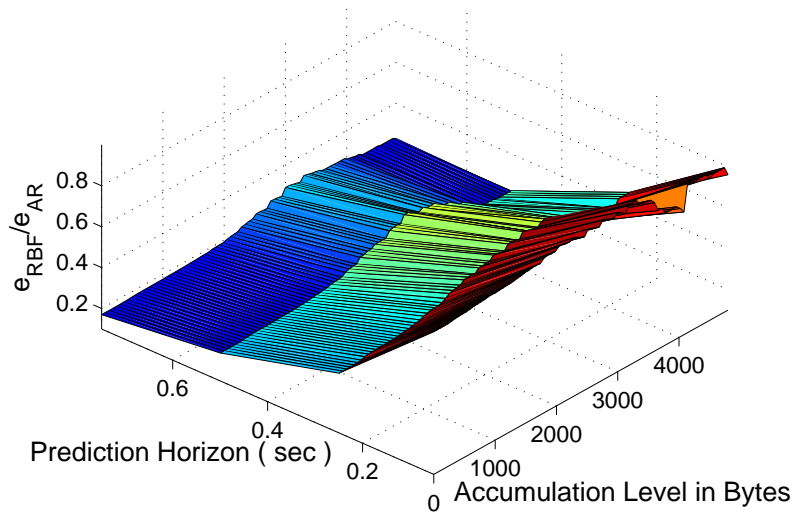


Fig. 60. Prediction of byte-accumulation with RBF and AR on path-1.

plot for SP, this implies that AR and ARMA still result in  $NSR$  just close to 1. ARX, ARMAX and RBF predictors show a very comparable prediction accuracy with RBF consistently performing slightly better than the linear predictors ARX and ARMAX. Moreover the ratio of errors of ARX, ARMAX and RBF predictions when compared to SP is close to 0.45 for 5 seconds ahead prediction. This implies that these predictors have an  $NSR$  less than 1. Hence, these three predictors are able to capture the effects of bit-rate fluctuation.

The predictors are then analyzed for the prediction accuracy with increasing accumulation levels. Figure 65 shows the relative performance of AR model when compared to Simple Predictor. It can be noticed that the relative performance of AR when compared with SP is almost constant beyond 3 seconds. Moreover there is no change in prediction accuracy with increasing accumulation levels. Figure 66

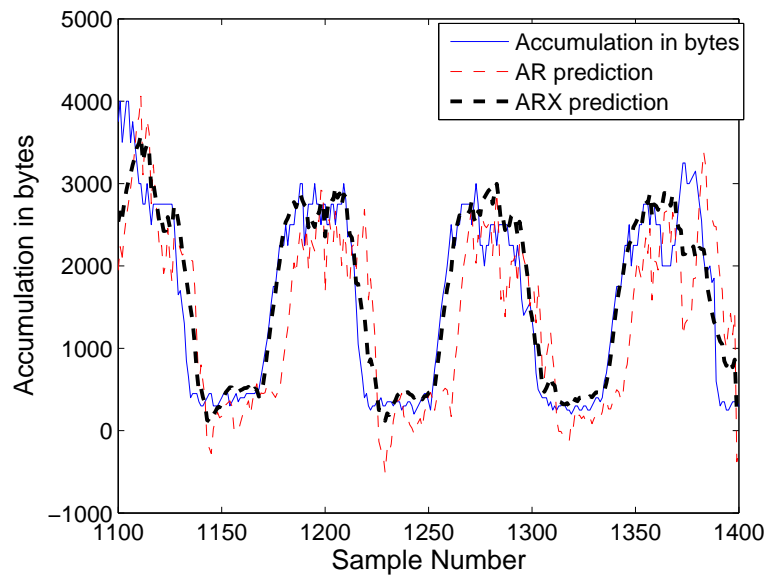


Fig. 61. 0.5 sec prediction of byte-accumulation with AR and ARX predictors on path-1.



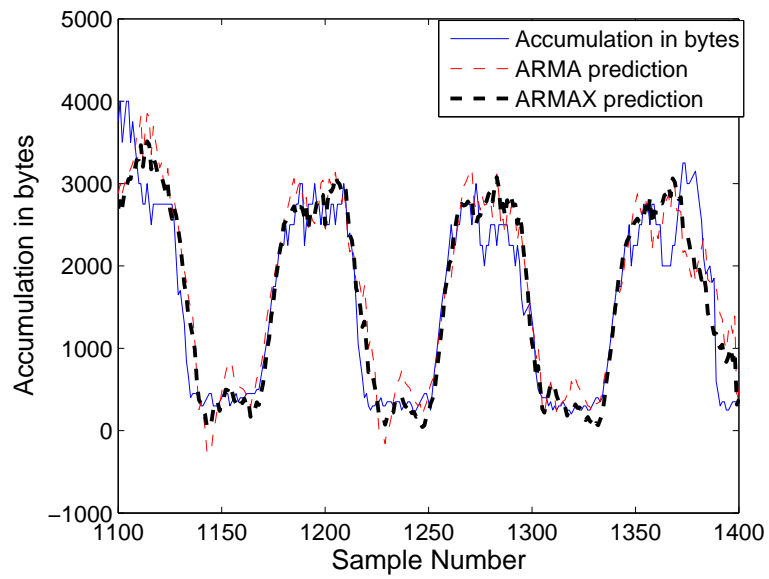


Fig. 62. 0.5 sec prediction of byte-accumulation with ARMA and ARMAX predictors on path-1.

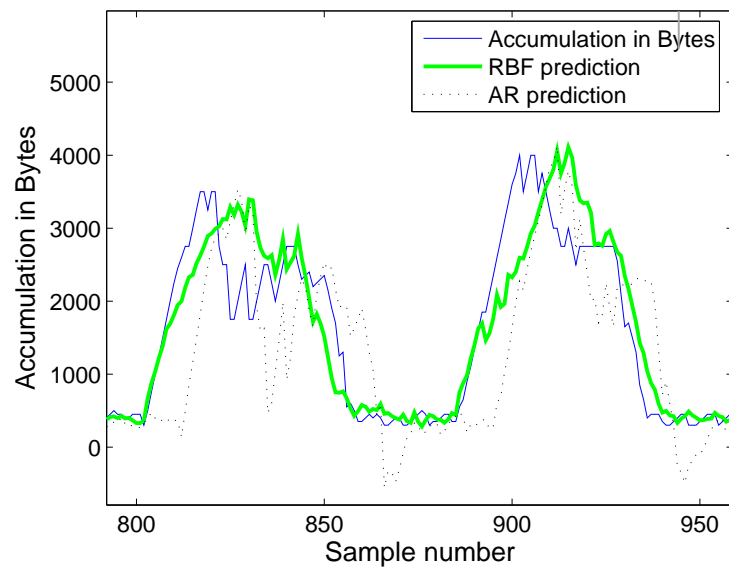


Fig. 63. 0.5 sec prediction of byte-accumulation with AR and RBF predictors on path-1.

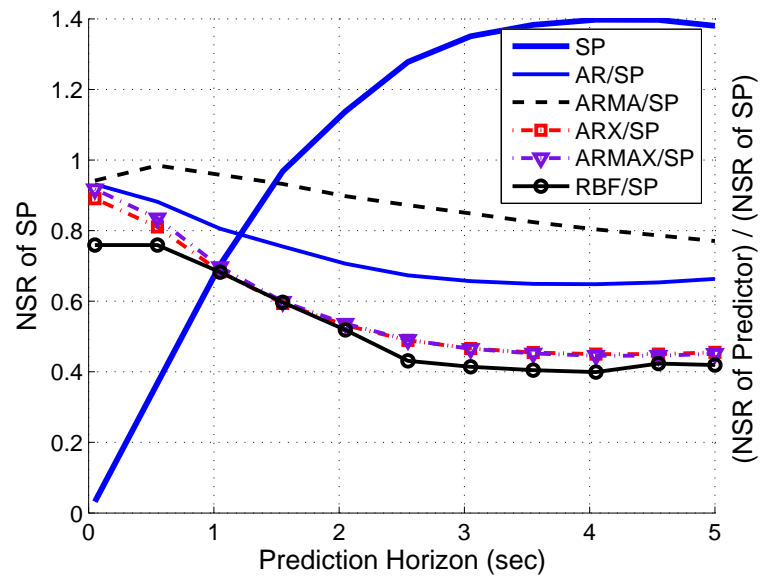


Fig. 64. Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for accumulation in bytes on path-3.

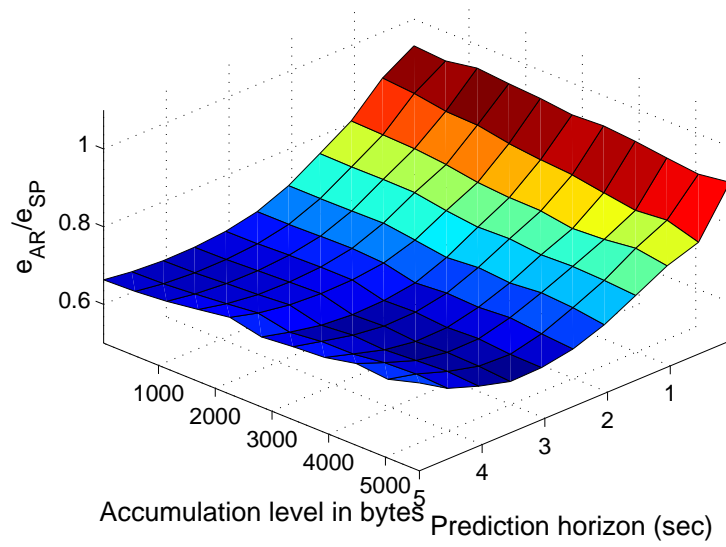


Fig. 65. Prediction of byte-accumulation with AR and SP on path-3.

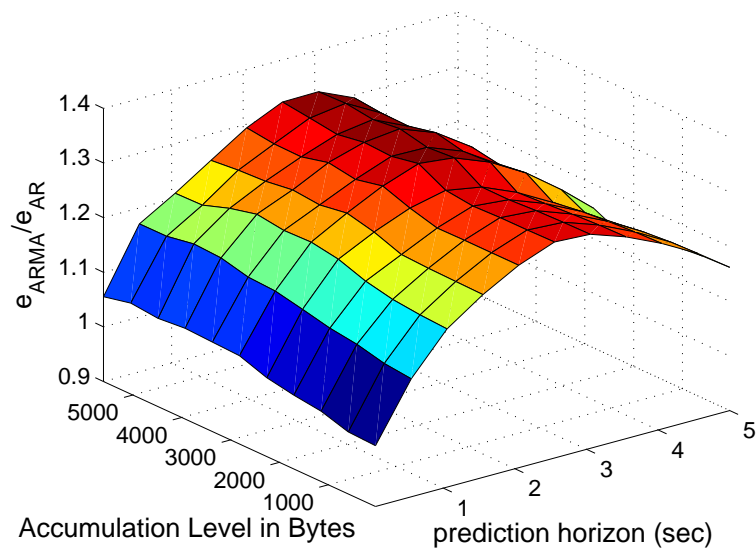


Fig. 66. Prediction of byte-accumulation with ARMA and AR on path-3

shows the comparison of ARMA and AR models. It can be noticed in this case that ARMA is worse off than AR predictor. Figure 67 and 68 show the prediction performance of ARX and ARMAX predictors when compared to AR model. We see that as the prediction horizon increases, the relative prediction accuracy increases. There seems to be no significant difference in the accuracy of the two predictors. The key observation is that the prediction accuracy improves beyond an accumulation level of 3000 bytes. Figure 69 shows the comparison of AR and RBF predictors. It can be noticed that RBF performs slightly worse than ARX or ARMAX at higher accumulation levels. Based on these observations, ARX is the best predictor of all the predictors investigated in this research.

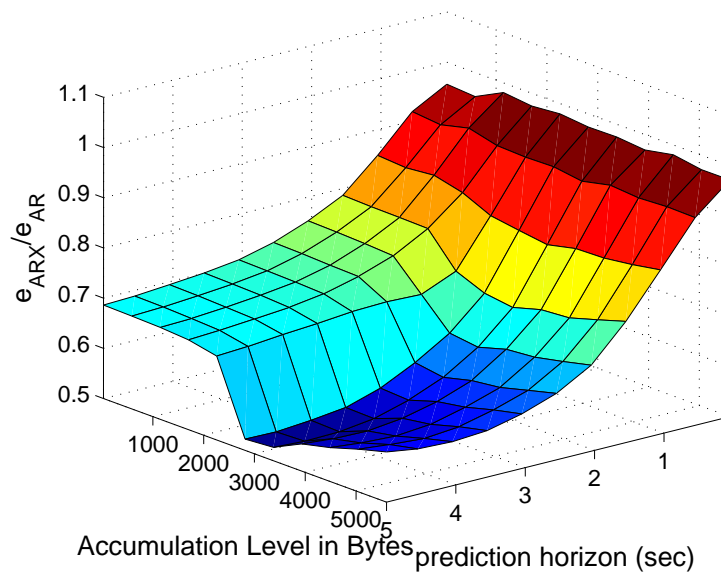


Fig. 67. Prediction of byte-accumulation with ARX and AR on path-3.

Figures 70 and 71 shows the prediction of AR, ARMA, ARX, ARMAX predictors. Figure 72 shows the predictions of RBF and AR predictors. The ARX and ARMAX

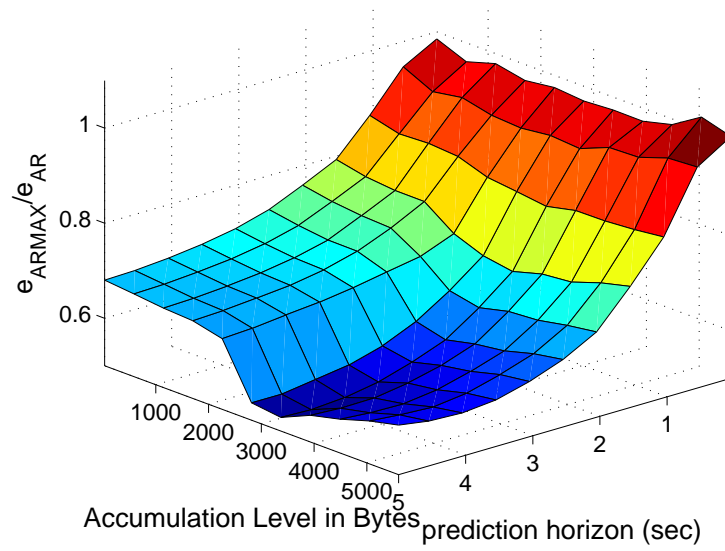


Fig. 68. Prediction of byte-accumulation with ARMAX and AR on path-3.

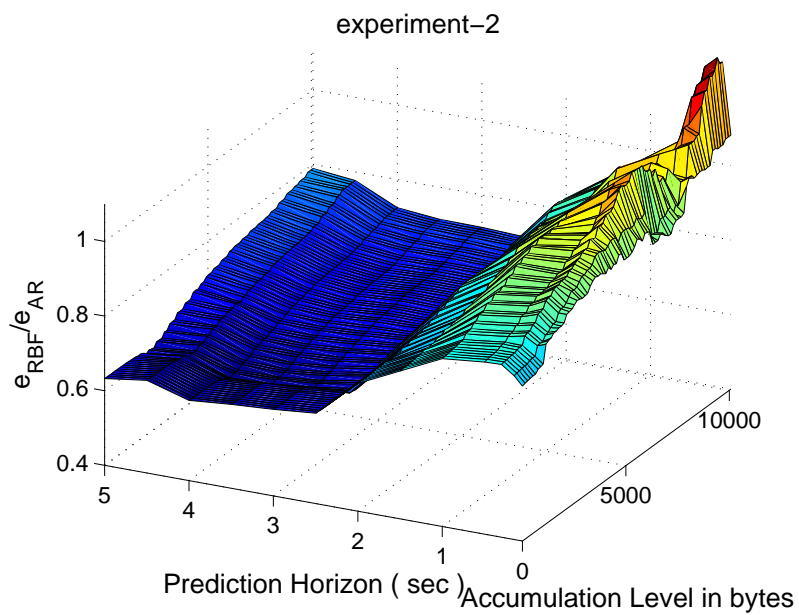


Fig. 69. Prediction of byte-accumulation with RBF and AR on path-3.

predictors are smoother than the RBF predictions. The AR prediction is completely out-of-phase with the accumulation signal.

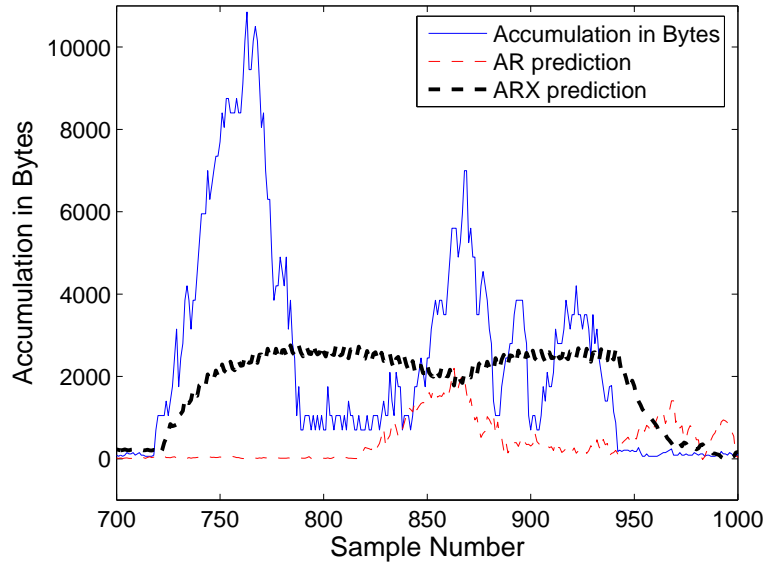


Fig. 70. 0.5 sec prediction of byte-accumulation with AR and ARX predictors on path-3.

### 3. Results for Prediction of Accumulation in Packets

#### a. Experiment 1

Figure 73 shows the  $NSR$  when using a Simple Predictor and also shows the ratio of errors when using AR, ARMA, ARX, ARMAX and RBF predictors as compared to SP with increasing prediction horizons. It can be noticed that the  $NSR$  when using SP increase linearly till 1 sec but the value is quite small (less than 0.2 for 1 sec prediction.) This implies SP is a useful predictor upto 1 second and beyond. It can be noticed that the AR and ARMA predictors reduce the error by only 0.9 times than when using SP. The ARX, ARMAX and RBF predictors reduce the prediction

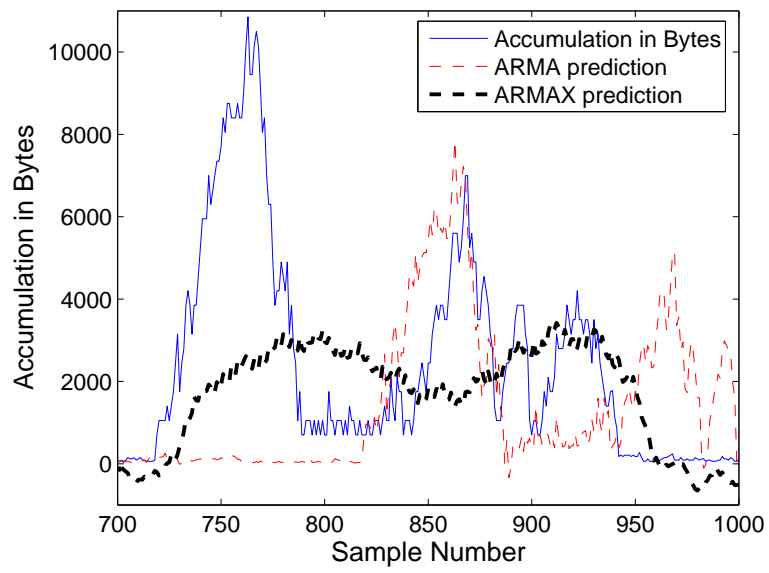


Fig. 71. 0.5 sec prediction of byte-accumulation with ARMA and ARMAX predictors on path-3.



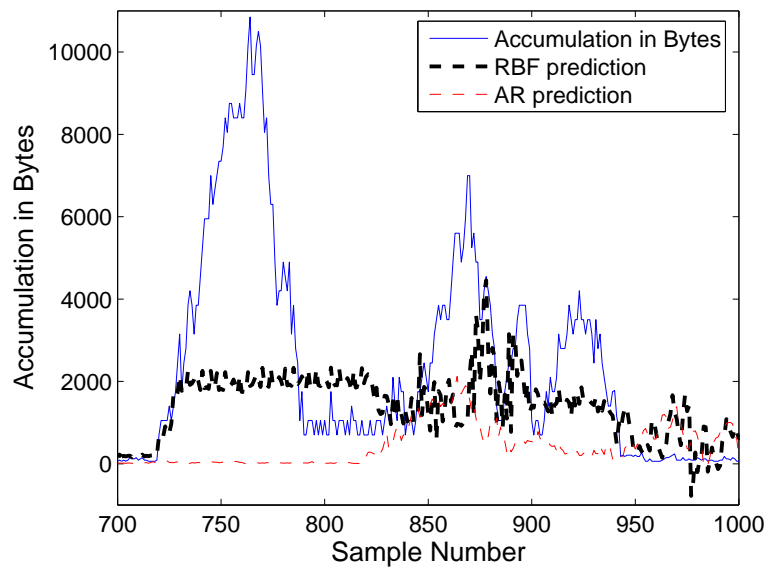


Fig. 72. 0.5 sec prediction of byte-accumulation with AR and RBF predictors on path-3.

error to about 0.4 times than when using SP. This implies that the ARX, ARMAX and RBF models (that include bit-rate as an input) are able to explain the effect of variation on input bit-rate on the accumulation signal. It can also be noticed that ARX, ARMAX and RBF predictors have a very comparable  $NSR$  values, though RBF performs slightly better than the two linear models.

The predictors are then analyzed for their prediction accuracy with increasing accumulation level. Figure 74 shows the relative performance of AR predictor when compared to SP. The prediction of AR predictor is quite comparable to that of SP. The error dropped by only 90%. Figure 75 shows the comparison of ARMA and AR models. ARMA performs only as good as AR predictor. Figures 76 and 77 show the comparison of ARX and ARMAX models with an AR model. It can be noticed that the performance improvement is quite substantial. The relative performance of ARX and ARMAX models improves with prediction horizon quite drastically. This indicates that only the long-term effect of change in bit-rate on packet loss can be captured by ARX model. Figure 78 shows the prediction of RBF predictor when compared to an AR predictor. It can be noticed that RBF relative performance increases with increasing prediction horizon. Moreover, the RBF network shows improvement in prediction accuracy with increasing accumulation levels. Therefore, RBF predictor is best suited in predicting the accumulation signal in packets for this experiment.

Figures 79 and 80 shows the prediction of packet accumulation using AR, ARMA, ARX, ARMAX predictors. The figure shows that the inclusion of bit-rate as an input in the ARX and ARMAX models improves the prediction accuracy quite appreciably. The figure 81 shows the time-series plot of RBF prediction and AR prediction.

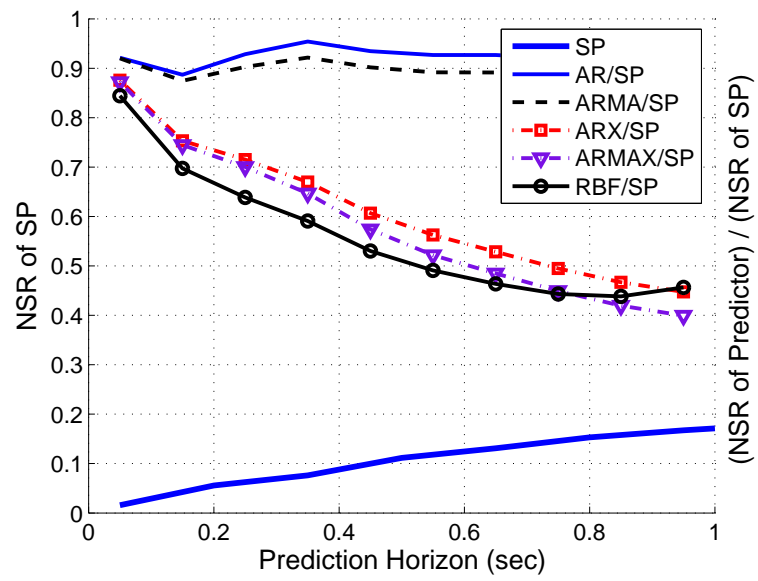


Fig. 73. Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon on path-1.

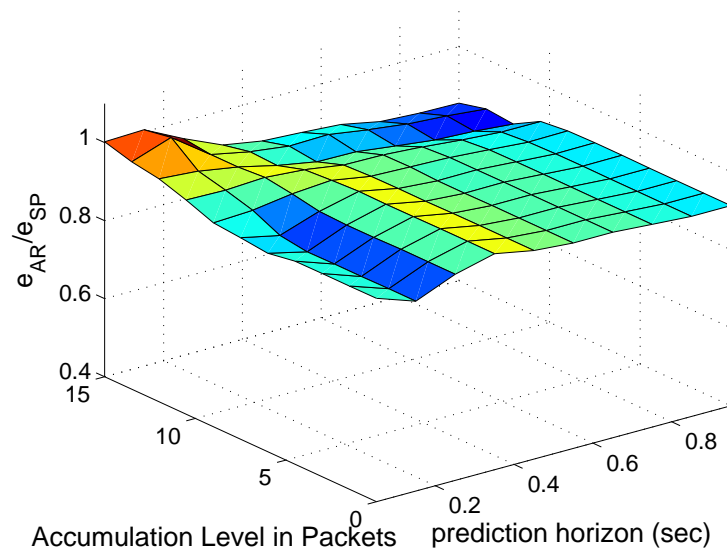


Fig. 74. Prediction of packet-accumulation with AR and SP on path-1.

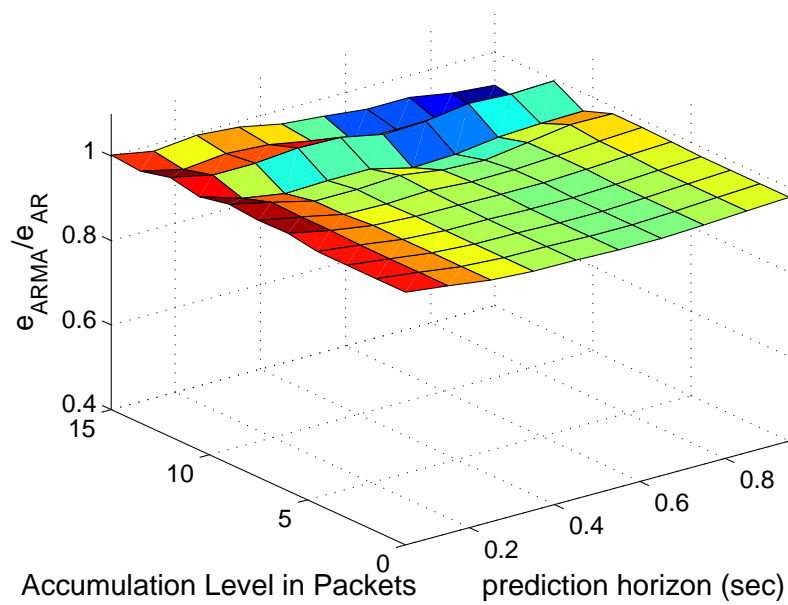


Fig. 75. Prediction of packet-accumulation with ARMA and AR on path-1.

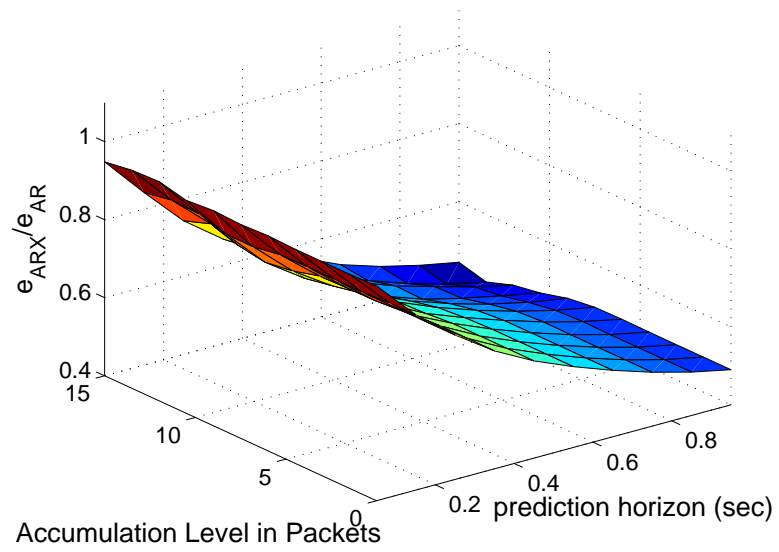


Fig. 76. Prediction of packet-accumulation with ARX and AR on path-1.

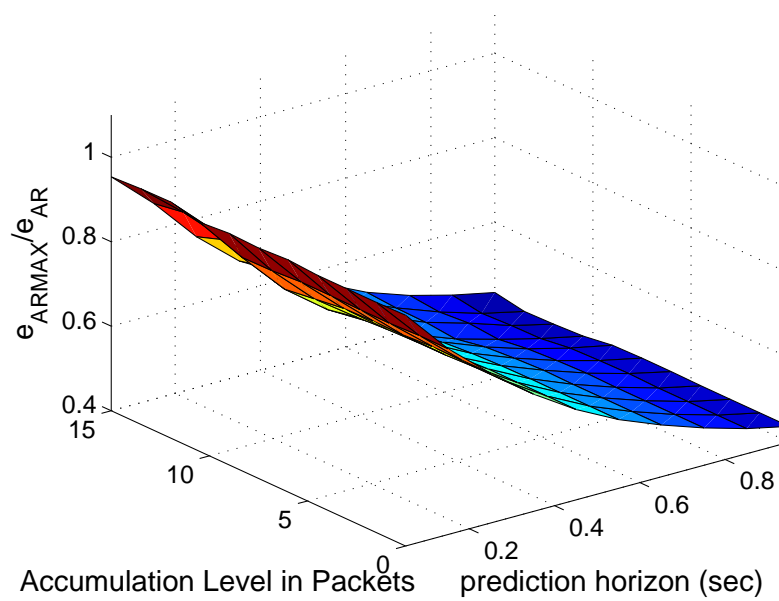


Fig. 77. Prediction of packet-accumulation with ARMAX and AR on path-1.

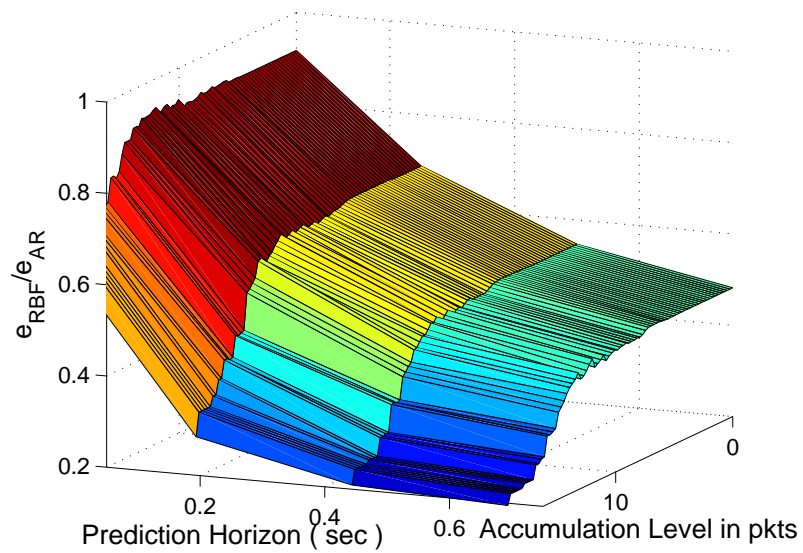


Fig. 78. Prediction of packet-accumulation with RBF and AR on path-1.

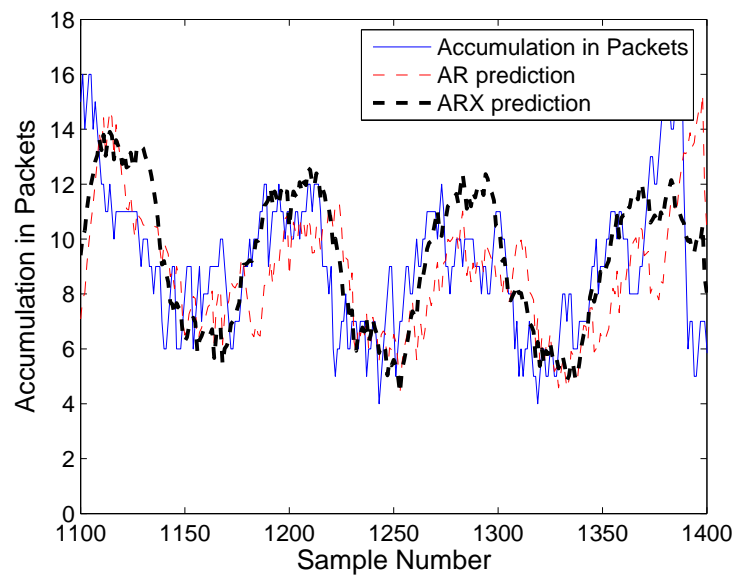


Fig. 79. 0.5 sec prediction of packet-accumulation with AR and ARX on path-1.

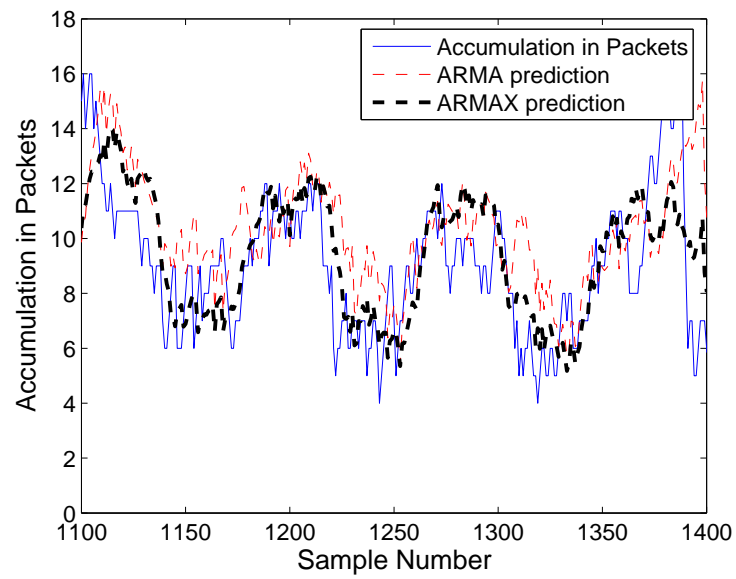


Fig. 80. 0.5 sec prediction of packet-accumulation with ARMA and ARMAX predictors on path-1.

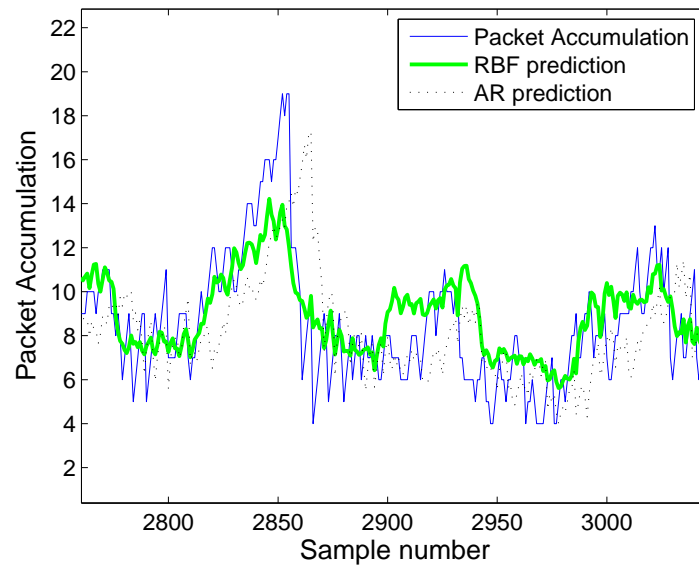


Fig. 81. 0.5 sec prediction of packet-accumulation with AR and RBF predictors on path-1.



### b. Experiments on Path-3

Figure 82 shows the *NSR* when using a Simple Predictor and also shows the ratio of errors when using AR, ARMA, ARX, ARMAX and RBF predictors as compared to SP with increasing prediction horizons. It can be noticed that the *NSR* when using SP increase linearly till 2 seconds and then stays close to 1 at higher prediction horizons. This implies that SP can be a useful predictor only upto prediction horizons of 1 seconds. It can also be noticed that the ARMA predictor is worse than SP. The AR predictor reduces the error by 75% at horizons greater than 2.5 seconds. The ARX and ARMAX predictors show a comparable performance and improvement when compared to the AR predictor. The RBF predictor is the best predictor of all the investigated predictors. The prediction error when using RBF settles down to about 50% (when compared to SP) at prediction horizons beyond 2 seconds.

The predictors are then analyzed for prediction accuracy with increasing accumulation levels. Figure 83 shows the performance of AR predictor when compared with a Simple Predictor (SP). It can be noticed that AR performs better than SP with increasing prediction horizon and with increasing accumulation levels. Figure 84 shows that ARMA results in a worse prediction than an AR model. Figures 85 and 86 show the comparison of ARX and ARMAX predictions with AR model. It can be noticed that ARX and ARMAX relative prediction accuracy is almost the same as AR model. The small improvement can be noticed at higher accumulation levels. Figure 87 shows the prediction performance of RBF network when compared to AR model. It shows that with increasing accumulation levels, the accuracy initially improves and then slightly degrades. Thus, in this case, ARX, ARMAX and RBF predictors have a very comparable prediction and any of them can be used.

Figures 88 and 89 show the prediction of AR, ARMA, ARX and ARMAX mod-

els. The figures reveal that all the predictors have similar prediction. Figure 90 shows the time-series plot of the two predictors. The ARX and ARMAX predictors result in smoother prediction when compared to RBF prediction. This suggests that ARMAX is the best predictor for this case. But it should be noted that none of the predictors for path-3 result in an  $NSR$  comparable to that obtained in the case of path-1. Thus bit-rate may not be a good control input in the case of path-3.

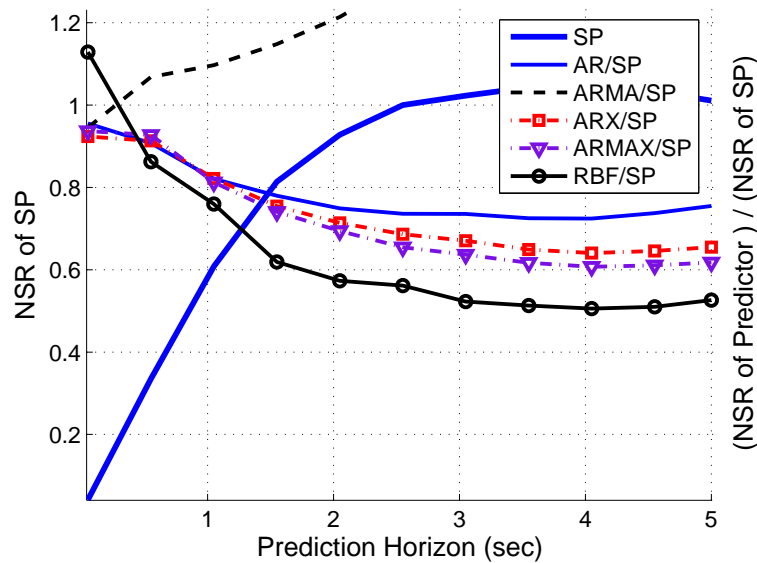


Fig. 82. Comparison of SP, AR, ARX, ARMAX and RBF predictors with increasing prediction horizon for accumulation in packets on path-3.

#### D. Chapter Summary

This chapter presented the effect of change in bit-rate on packet loss and end-to-end delay. Experiments were conducted on two paths; a domestic path and an international path. The international path had high end-to-end delay and experienced higher congestion when compared to the domestic path. The experimental data was tested

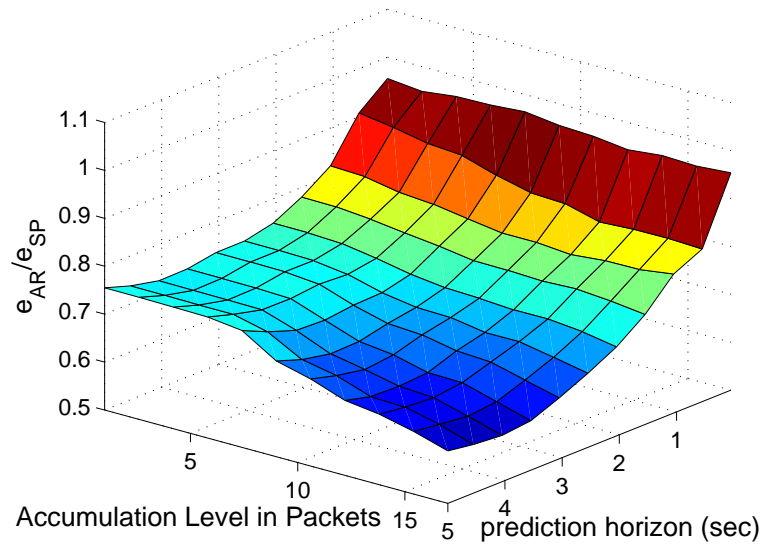


Fig. 83. Prediction of packet-accumulation with AR and SP on path-3.

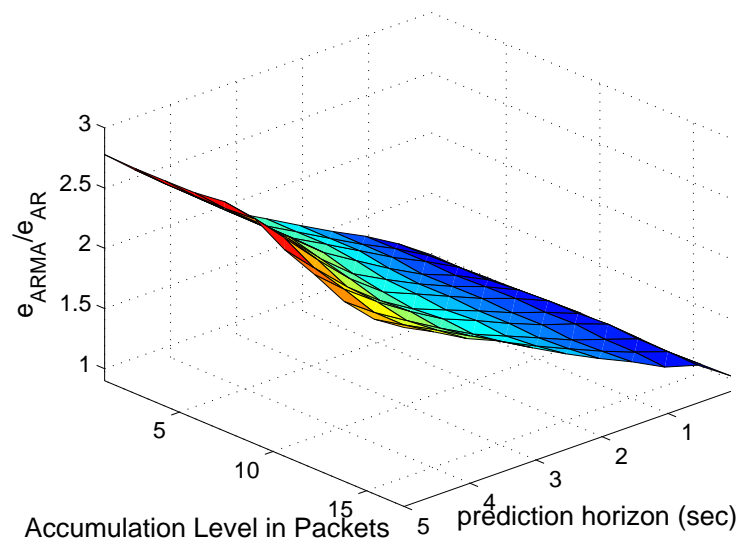


Fig. 84. Prediction of packet-accumulation with ARMA and AR on path-3.

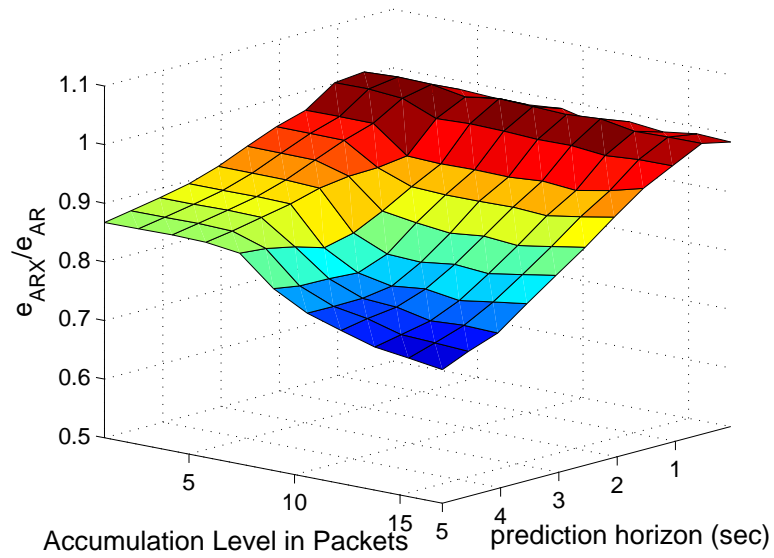


Fig. 85. Prediction of packet-accumulation with ARX and AR on path-3.

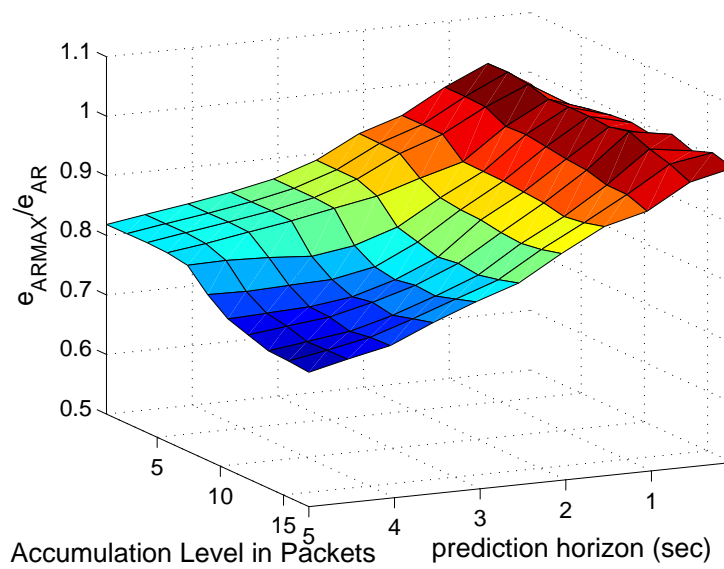


Fig. 86. Prediction of packet-accumulation with ARMAX and AR on path-3.

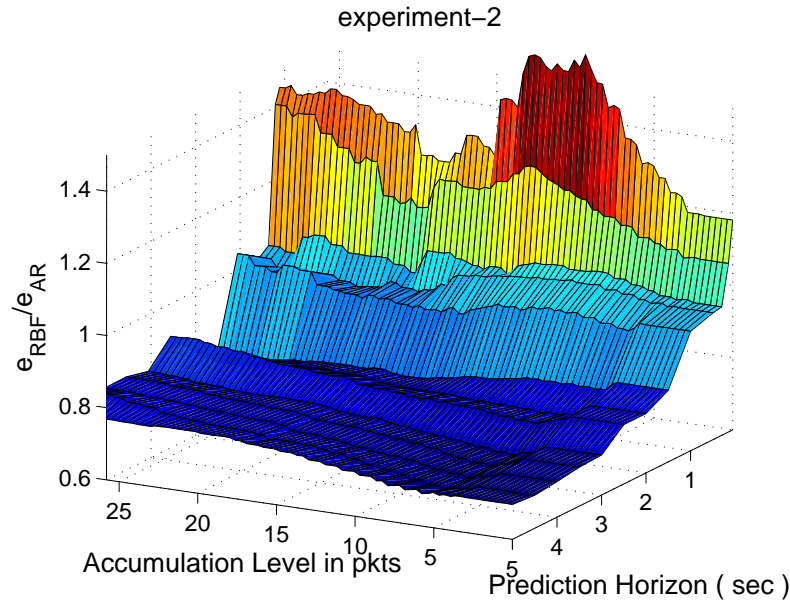


Fig. 87. Prediction of packet-accumulation with RBF and AR on path-3.

for statistical differences for different input bit-rates. The accumulation signal was also analyzed for predictability when accumulation is in terms of packets and bytes.

The statistical experiments of path-1 data show that the effect of change in bit-rate is more when compared to the path-3. The above fact can be verified by comparing figures 42 and 47. The time-ensemble average revealed a settling time close to 1 second for path-1 and about 5 seconds for path-3. Therefore, the design of any control algorithm which uses bit-rate as an input should be aware of this inherent slack in the system response.

When the accumulation signal is measured in bytes, path-1 and path-3 reveal some similarities as seen from Figures 55 and 64. In both the cases, the *NSR* of the prediction increases linearly with the prediction horizon. In the case of path-1, the *NSR* of SP is about 0.7 at 1 second and in the case of path-3 it crosses crosses 1 at

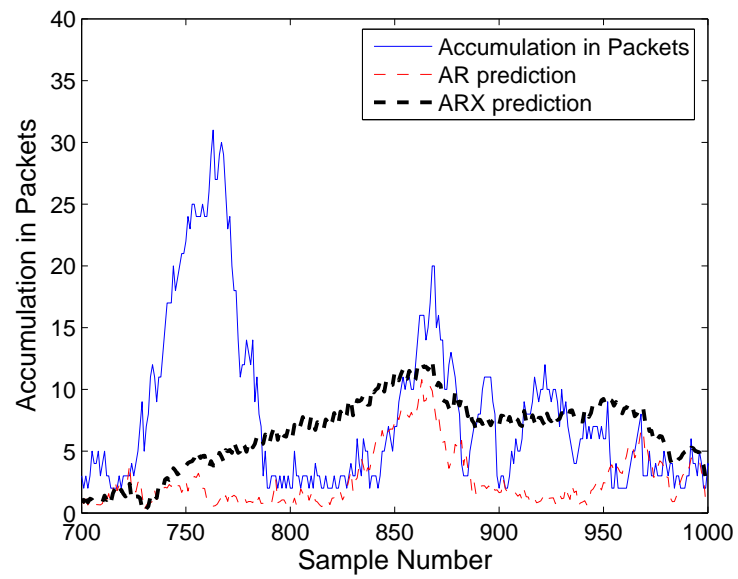


Fig. 88. 0.5 sec prediction of packet-accumulation with AR and ARX models on path-3.

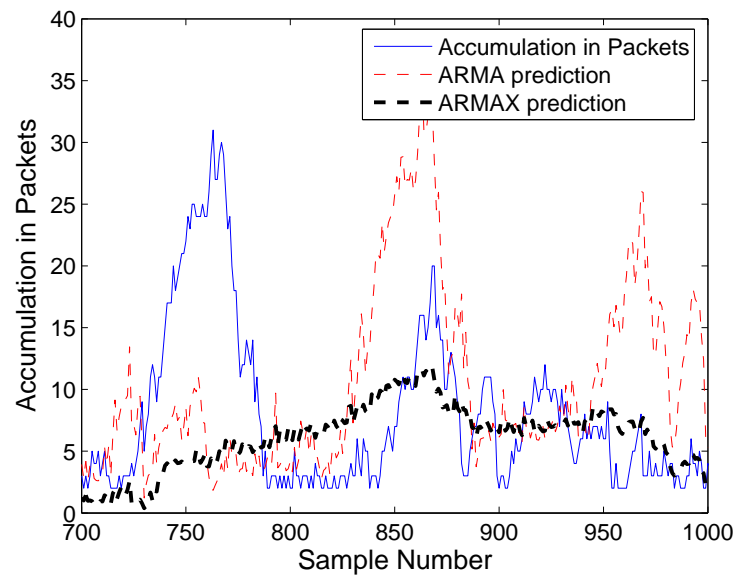


Fig. 89. 0.5 sec prediction of packet-accumulation with ARMA and ARMAX models on path-3.

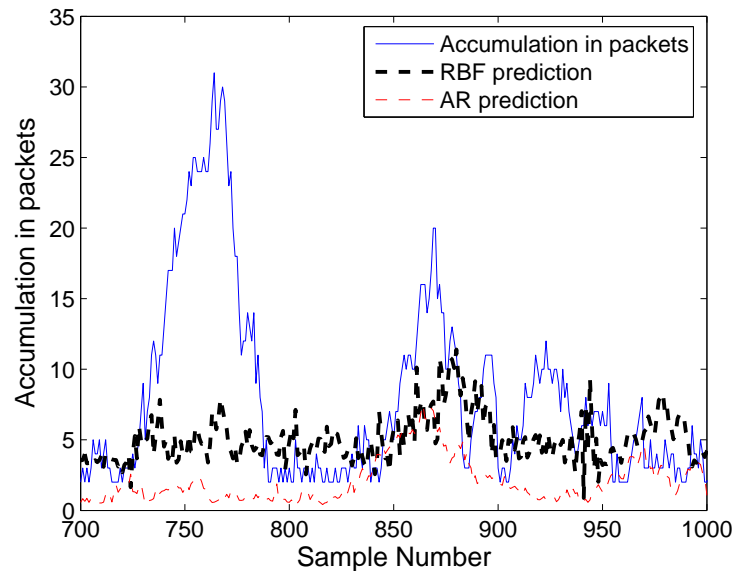


Fig. 90. 0.5 sec prediction of packet-accumulation with RBF and AR on path-3.

1.5 seconds. This indicates that SP is not useful when prediction horizons beyond 1.5 second are required. The difference between path-1 and path-3 are quite evident when the predictive models are compared. Though the AR model does not perform much better than SP in both the cases, ARMA performs remarkably better in the case of path-1. The differences are even more visible when ARX, ARMAX and RBF predictors are used. In the case of path-1, the above three predictor reduce the error to 10% of SP error where as in the case of path-3, the reduction is only to about 65% at 1 second and close to 40% at 5 seconds prediction. Therefore, the predictive models could not capture the effects of bit-rate variation on the accumulation signal in bytes in the case of path-3. In both the cases, RBF predictor resulted in lower  $NSR$  value, closely followed by ARX and ARMAX predictors. There was essentially little difference in the performance between ARX and ARMAX predictors. ARMAX



has a disadvantage in terms of high computational complexity. Based on the comparative analysis of the predictors for increasing accumulation levels, ARX was the most suitable predictor for both the experiments.

When the accumulation signal is measured in packets, path-1 and path-3 reveal greater contrast as can be seen from Figures 73 and 82. The *NSR* of the SP is as low as 0.2 for path-1. In the case of path-3, *NSR* of the SP increased linearly up to 1.5 seconds and settled at 1 at 2.5 seconds. Thus, SP is useful for more than 1 second prediction for path-1. In the case of path-3, SP has meaning till about 2.5 seconds. The predictive models AR, ARX, ARMAX and RBFs developed in this research have an *NSR* less than 1 for longer prediction horizons when compared to SP. The time-series predictors AR and ARMA show only a slight improvement over SP. In the case of path-3, ARMA is actually worse off than SP. The ARX and ARMAX predictors have a very comparable performance. RBF predictors gave a better performance than ARX in the case of path-3. When the predictors were compared for increasing accumulation levels, RBF predictor turned out to be the best choice. The prediction of accumulation signal in packets supports the statistical differences observed in this research. In the case of path-1, increasing the bit-rate by a factor of two showed significant effect on the end-to-end delay and total losses. The ARX, ARMAX and RBF models using bit-rate as one of the modeling inputs showed a remarkable improvement over AR and ARMA models (that do not include bit-rate as an input.) In the case of path-3, the ARX, ARMAX and RBF models showed only a small improvement over AR and ARMA predictions. This result is in agreement with the observation that switching the bit-rate by a factor of five showed only a minor statistical effect on the end-to-end delay or on the total losses. But in all the cases, the developed models ARX, ARMAX and RBF had an *NSR* value less than 1 for longer prediction horizons when compared to SP. This suggests that

predictive control algorithms based on these models have a good hope for improving the end-to-end characteristics of real-time applications.

## CHAPTER V

## SUMMARY AND CONCLUSIONS

## A. Summary

The objective of the current research was to develop prediction techniques suitable for controlling real-time applications across a Wide-Area-Networks (WAN). Accumulation of bytes in the network was chosen to represent the end-to-end single flow characteristics. Various linear and non-linear models were proposed to capture the above mentioned accumulation signal. The linear techniques include Auto-Regressive (AR), Auto-Regressive Moving Average (ARMA), Auto-Regressive Exogenous (ARX) and Auto-Regressive Moving Average Exogenous (ARMAX) models. Radial Basis Functions and Sparse Basis Functions were the non-linear models investigated. The accumulation signal was analyzed for two cases. In the first case, the bit-rate was kept constant and the network was sampled at regular intervals using UDP probe packets. This helped us in evaluating the capacity of the models in capturing the characteristics of cross-traffic. In the second case, the input bit-rate was varied as a square wave to statistically analyze the effect of input bit-rate on end-to-end delay and packet loss. Linear and non-linear models were then used to analyze which modeling technique best explain the statistical results. All the predictors were analyzed for different accumulation levels and for increasing prediction horizon. A baseline predictor called the ‘Simple Predictor’ was used to analyze the relative performance of all the predictors.

Chapter I presented the motivation for developing predictive techniques in implementing end-to-end control for real-time applications. A review of contemporary literature on end-to-end measurement and analysis of Wide-Area-Network was first

presented. It was followed by a discussion on the failure of network simulators in capturing real world network. An overview on the application of System Identification techniques, neural networks and fuzzy to predict the network characteristics is then presented. This was followed by a discussion on the motivation for using accumulation signal as an indicator of the network state. Finally the proposed approach of the current research was presented.

Chapter II first discussed the broad categories of modeling techniques. It was followed by a brief description of the key steps involved in identifying a system. A brief overview of the linear and non-linear modeling techniques along with their respective advantages and disadvantages was then presented. The linear models discussed were AR, ARMA, ARX and ARMAX. The non-linear models discussed were Radial Basis Functions and Sparse Basis Functions. All the predictors were to be compared with a baseline predictor called the ‘Simple Predictor’.

Chapter III gave a brief description of the experimental set-up used in this research followed by the time-series modelling results. Simulations were conducted on a network simulator called ns-2. The simulation cross-traffic is ensured to match the real-world scenario as closely as possible. The real-world experiments were conducted on an overlay network called PlanetLab. The auto-correlation plots of the three time series was first presented. It was concluded that the auto-correlation of ns-2 simulated data decreased more slowly when compared to the real world traces. The second distinctive feature was that for the simulated time-series, the autocorrelation would drop to zero and then stay very close to zero. But for real world time-series, the auto-correlation would have a steady value of about 0.1 even after 20 sec duration. This indicates that real-world traces may have long term dependency that is not captured by simulations. The linear AR predictor was developed and compared with a ‘Simple Predictor’ (SP) for prediction horizons varying between 50ms to 2 sec

and for increasing accumulation levels. It was observed that the AR model is as good as SP at shorter prediction horizons. But with increasing prediction horizon, the AR prediction error drops down to about 70-80% of SP prediction error. The second feature to be noticed is that there does not seem to be any advantage with increasing accumulation levels.

Radial Basis Functions (RBF) based predictors were then developed. The most important degrees of freedom of RBF that were analyzed were the radius of the basis function and the number of neurons. A larger radius of the basis function resulted in smoother prediction and better generalization. But the tradeoff was in terms of reduced ability to specialize in detecting the peaks. Larger radius made the predictor more conservative in predicting the peak in the accumulation signal. Separate training, test and validation data sets were used to avoid the drawbacks of over-training the network. The order of the radius of the turned out to be the order of the norm-2 of the distance of the input vectors. The number of neurons was close to 50 in almost all the cases. The performance of RBF was then compared to that of AR. The *NSR* of RBF models was found to be less than that of AR of SBF models in all the cases. On simulated data, it was found that though the overall prediction of both the predictors was comparable, RBF predicts the peaks in accumulation better than AR. The RBF prediction error was 80% that of AR error. On path-1, RBF provided better prediction accuracy with increasing prediction horizon and with increasing accumulation levels. The error could be reduced by about 80% that of AR error. On 'path-2', two models of RBF predictor were developed: RBF1 and RBF2. RBF1 used only 10 inputs (of 0.5 seconds) where as RBF2 used the 50 inputs, with about 40 inputs of 17 seconds old in addition to the 10 inputs of 0.5 seconds old. RBF1 gave a prediction accuracy slightly worse than AR model. AR needed close to 17 seconds of data where as RBF1 used only 0.5 seconds of old data to give a

comparable prediction. If an AR model using only 0.5 sec was developed, the RBF network could give an error of about 70% the error of the AR model. This feature indicated that RBF1 acted as an early congestion detection scheme. RBF2 was found to give the least *NSR* of all the predictors but was found to fail in predicting some of the peaks in the accumulation signal.

The key parameters to tune a Sparse Basis Function (SBF) were the forgetting factor, the number of initialization points, the number of basis functions in the dictionary of functions, the number of basis functions selected and the number of backward forward search iterations. From experiments it was found that the number of backward forward iterations of less than 10 was sufficient in almost all the cases. The dictionary of basis functions was chosen to be close to 60 and included many linear and non-linear transformations on the input vectors. The selected basis functions was kept at 5. Increasing the selected basis to 10 was found to result in instability. Of the two remaining parameters the forgetting factor was more important and experiments showed that the error was linearly dependant on it. A value very close to 1 (0.99) was found to be a good choice. The number of initialization points close to 400 was found to be sufficient for path-2. This actually corresponded to the peak in the auto-correlation function for the time-series of the signal. The prediction accuracy of Sparse Basis (SBF) was compared to that of Simple Predictor (SP). SBF was worse off than SP at increasing accumulation levels and at increasing prediction horizon. It was therefore concluded that SBF is not a good predictor for the accumulation signal. There were other problems with Sparse Basis functions. Sparse basis could be efficiently and recursively implemented for single step prediction but a multi-step prediction was difficult to implement. The second disadvantage was that including an input (as in ARX) was not straightforward.

In conclusion, the RBF predictor can be tuned to give the least *NSR* of all the

three predictors tested in this research. But high accumulation levels is the region where the accuracy of the predictor is very crucial. Therefore, the accuracy of the predictors was also examined with increasing accumulation level. The results show that in the case of ns-2 simulation and path-1, RBF predictors have a slight advantage with increasing accumulation levels. In the case of path-2, AR model of order 350 (using 17 seconds) was found to give the best prediction. But if computational complexity prohibits developing or using such a high order AR models, an RBF model with just 10 inputs (0.5 seconds) was found to give a comparable prediction and can be treated as the next best alternative.

Chapter IV analyzed the effect of input bit-rate on packet loss and end-to-end delay. In real-time applications, a packet arriving after its deadline is equivalent to being lost. Based on this notion, the term ‘total loss’ is used to indicated the sum of packet loss and packets that arrive after the deadline. In end-to-end control of networked applications, the most popular input is the source bit-rate. Therefore, this chapter analyzed how bit-rate effected total loss. The experiments were conducted on two paths (path-1 and path-3) with high (300 ms) and low (100 ms) average one-way delay. The bit-rate was varied as a square wave between a low and a high value. The switching time was chosen low enough to ensure that the network conditions do not appreciably change within one time-period. The collected data was analyzed for statistical differences in total losses for the two bit-rates and for the dynamics of the losses. It was found that the bit-rate effects were more explicit on path-1 than on path-3. The second observation was that to observe any statistical differences, the bit-rate had to be only doubled for path-1 where as on path-3, the bit-rate had to be increased by a factor of five. This implied that the average one-way delay indicated the level of congestion and the amount of available bandwidth on a given path. Therefore, the effects of increasing bit-rate were more predominant on path-

1 and bit-rate is a promising control input for path-1 path. The second part of the chapter is dedicated to analyzing the extent to which the linear and non-linear models can capture and explain the effects of input bit-rate on the accumulation signal. The accumulation signal was calculated in two different ways. First, accumulation in bytes was analyzed and then accumulation of packets was examined. Accumulation in bytes was the favored choice for control applications because it facilitates the direct inclusion of bit-rate as an input. The models developed were AR, ARMA, ARX, ARMAX and RBF. AR and ARMA were time-series models and do not take the input bit-rate changes into account. The ARX, ARMAX and RBF include bit-rate as in input. For path-1, the ARX, ARMAX, and RBF models outperform the AR and ARMA models. The improvement is predominant at higher prediction horizons. RBF performance is only slightly better than ARX or ARMAX. This is not the case of path-3. Here all the models have comparable performance except for RBF network, which shows a marked improvement over AR models. The accumulation signal was then calculated in terms of the number of packets. For path-1, ARX and ARMAX models resulted in prediction error of 50% that of AR models but an RBF network could give prediction with about 60% that of AR model. But for path-3, ARX, ARMAX and RBF could not show any considerable improvement over a time-series based model. This implies that for paths with low average delays, time-series models should be used and paths with high average delay can benefit from the inclusion of bit-rate as an input. Though the prediction accuracy of RBF and ARX predictors is not very good on path-3 compared with path-1, they still show considerable improvement over SP. In the case of path-3, the *NSR* of SP crosses 1 at about 2 seconds, but the *NSR* of 5 second prediction horizon using RBF and ARX predictors is about 0.5 for accumulation in packets and about 0.7 for accumulation in bytes. Thus the developed predictor have extended the predictable horizon when compared to SP. Control algorithms such as



Model Predictive Control (MPC) are effective only to the extent to which the models are able to predict the output. Since the models developed in this research have improved the prediction of the accumulation when compared to SP, MPC techniques are promising control strategies for real-time control over the Internet.

## B. Conclusions and Recommendations

The following are the conclusions drawn from this study:

1. Linear system identification techniques (AR, ARMA, ARX, ARMAX) and Radial Basis Functions (RBF) are promising techniques in modeling end-to-end characteristics of Wide-Area Networks. Sparse Basis Functions are not very suitable for the above purpose.
2. The RBF networks capture the congestion duration better than the linear models.
3. On paths with high end-to-end delay (path-1), bit-rate is a very promising control input. The accumulation signal on path-1 is also quite predictable. One second prediction using ARX and RBF predictors give an *NSR* is close to 0.1 for accumulation in bytes and close to 0.5 for accumulation in packets.
4. On paths with low end-to-end delay (path-3), bit-rate may not be the best control input. The statistical analysis on ‘total loss’ indicate that increasing the bit-rate by eight folds shows only a small increase in ‘total loss’. The primary reason may be that the path is very well provisioned and hence, sensitivity to input bit-rate is very low. Therefore, new control inputs or techniques need to be investigated.

5. ARX and RBF models extend the predictable horizon of the accumulation signal when compared to SP on all the paths investigated (path-1 and path-3). For path-3, *NSR* of SP crosses 1 at 2 seconds but with ARX and RBF predictors, even at 5 seconds, the *NSR* is close to 0.7 for accumulation in bytes and close to 0.5 for accumulation in packets.

The following recommendations are proposed for further research in this area:

1. Since RBF models give better prediction during congestion, there seems to be a certain non-linearity that affects the accumulation signal that is not being captured adequately by the linear models. Therefore, intelligent transformation of the accumulation signal to eliminate this non-linearity is very desirable.
2. Predictive control algorithm should be developed over paths with high average delay.
3. For paths with low average delay, new intelligent schemes based on time-series models should be developed. One of the promising avenues is the use of multiple paths between the source and destination. The control algorithm then becomes switching judiciously between the two or more paths based on time-series prediction of the accumulation signal on each path.
4. Non-real-time applications can also benefit from the current research if the objective is to optimize the throughput of the application. In the case of corporate leased lines where the corporation is charged for total bits transferred, time-series predictions can be used to hold the source from sending data when it predicts very high accumulation signal.

## REFERENCES

- [1] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 458-472, August 1999.
- [2] E.F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. New York: Springer 2004.
- [3] A. Bhattacharya, A. G. Parlos and A. F. Atiya, "Prediction of MPEG-coded video source traffic using recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2177-2190, August 2003.
- [4] J.-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proceedings of SIGCOM*, vol. 23, no. 4, October 1993, pp. 289-298.
- [5] V. Paxson, "End-to-end internet packet dynamics," in *Proceedings of ACM SIGCOMM*, vol. 7, no. 3, 1997 pp. 139-152.
- [6] C.J. Bovy, H.T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal and P. V. Mieghem, "Analysis of end-to end delay measurements in the Internet," in *Proceedings of the PAM 2002 Conference*, Fort Collins, CO, March 25-26 2002, pp. 139-148.
- [7] "Network simulator 2 (ns-2)," Information Science Institute, University of Southern California, Los Angeles, 2002.
- [8] V. Paxson and S. Floyd, "Why we don't know how to simulate the Internet," in *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, GA, December 1997, pp. 1037-1044.

- [9] A. Sang and S-q Li, "A predictability analysis of network traffic," in *Proceedings of IEEE INFOCOM 2000*, vol.39, no. 4, July 2002, pp. 329-345.
- [10] K. Shah, S. Bohacek and E. Jonckheere, "On predictability of data traffic," in *Proceedings of IEEE American Controls Conference*, Denver, CO, June 2003, pp. 1619-1624.
- [11] H. Ohsaki, M. Murata and H. Miyahara, "Modeling end-to-end packet delay dynamics of the Internet using system identification," in *Proceedings of 17th International Teletraffic Congress*, Salvador, Brazil, December 2001, pp. 1027-1038.
- [12] H. Ohsaki, M. Morita and M. Murata, "On modeling round-trip time dynamics of the Internet using system identification," *IEICE Transactions on Communications*, vol.E85-B, no. 1, January 2002, pp. 11-19.
- [13] C. You and K. Chandra, "Time series models for Internet data traffic," in *Proceedings of 24th Conference on Local Computer Networks, LCN-99*, Lowell, MA, October 1999, pp. 164-171.
- [14] A. G. Parlos, "Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol.3, no. 3, May 2002, pp. 2460-2465.
- [15] Q. P. Wang, D. L. Tan, N. Xi and Y. C. Wang, "The control oriented QoS: Analysis and prediction" in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 2001, pp. 1897-1902.
- [16] Q. Jiang, R. Srinivasan and D. Slonowsky, "Measurement based traffic prediction using fuzzy logic," in *IEEE CCECE Canadian Conference on Electrical and*

- Computer Engineering*, August 2002, vol.2, pp. 834-840.
- [17] S. Doddi, "Empirical modeling of end-to-end delay dynamics in best-effort networks," M.S. Thesis, Texas A&M University, College Station, TX, December 2003.
- [18] E. S. Yu and C.Y. R. Chen, "Traffic prediction using neural networks," in *Global Telecommunications Conference, 1993*, May 1993, vol.2, pp. 991-995.
- [19] Y. Xia, D. Harrison, S. Kalyanaraman, K. Ramachandran and A. Venkatesan, "An accumulation-based congestion control model," in *IEEE International Conference on Communications*, Anchorage, AK, May 2003, pp. 657-663.
- [20] V. Khariwal, "Adaptive control of media applications in best-effort networks," M.S. Thesis, Texas A&M University, College Station, TX, August 2004.
- [21] A. Konstantinou, "Flow control techniques for real-time media applications in best-effort networks," M.S. Thesis, Texas A&M University, College Station, TX, December 2004.
- [22] PlanetLab Available via <http://www.planet-lab.org> [Accessed January 2005]
- [23] A. F. Atiya, M. A. Aly and A. G. Parlos, "Sparse basis selection: New results and application to adaptive prediction of video source traffic," *IEEE Transactions on Neural Networks, Special Issue on Adaptive Learning Systems in Communication Networks*, vol.2, pp. 834-840, September 2005.

## VITA

The author, Mallik Kommaraju, was born on November 30, 1981 in Hyderabad, India. He received his Bachelor of Technology degree in mechanical engineering from Indian Institute of Technology, Madras, India in May 2003. In Fall 2003 he began the Master of Science degree program in the Mechanical Engineering Department, Texas A&M University, College Station and received his degree in December 2005.

## Permanent Address:

Mallik Kommaraju  
C-15, TTS, NTPC,  
Ramagundam, Andhra Pradesh, INDIA-505215

Phone: (091) 872-8272527

Email: mallik@neo.tamu.edu