Forecasting Heartbeat Delay for Failure Detection over Internet using Nonlinear System

Zhao Hai-jun[1,3], Ma Yan[1,2,3], Huang Xiao-hong[1,3], Zhao Fang[1,3]
[1]*State Key Laboratory of Networking and Switching Technology*
[2]*Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia*
[3]*Network Technology Research Institute*
Beijing University of Posts and Telecommunications, Beijing, 100876, China

Abstract

To overcome Internet dynamic characteristics and accurately predict next heartbeat message delay for failure detection service, a novel learning machine is proposed to predict next heartbeat arrival time. We use a nonlinear autoregressive network with exogenous inputs to learn nonlinear and linear characters of heartbeat messages, perform one-step-ahead prediction to estimate future heartbeat delay. The inputs are two moving window observations of past heartbeat delays and heartbeat sending time, the output is next heartbeat delay, the network is trained by standard back-propagation algorithm, its weights and basis are adjusted by approximate steepest descent rule. Simulation result shows that this adaptive algorithm can accurately capture heartbeat dynamics over internet and make minimum prediction error under different network environments such as bottleneck link, link down and up.

1. Introduction

Failure detection service was a well-known fundamental building block for ensuring fault tolerance in distributed systems [1]. Many implementations of failure detector were generally based on heartbeat messages, the accuracy of failure detection result depends on predicting heartbeat message arrival time [2]. However, the Internet transmits data packet with best effort with dynamic behavior [3] and provides no guarantee on the end-to-end transmission delay, which make it very difficult to accurately predict heartbeat arrival time.

The observations of heartbeat arrival time can be treated as a time series, many traditional model-based methods were used to model and predict next heartbeat message arrival time with its past observations , but which are based on the assumption -there is linear correlation between heartbeat arrival time. However, the real heartbeat arrival over internet behaves time-varying and non-linear. Above traditional models can provide reasonable results for linear or nearly linear time-invariant system, but might result in poor performance for cases in which the dynamics of the systems are highly nonlinear. Due to its strong approximation capability as well as its inherent adaptive learning ability, artificial neural network has been applied for nonlinear system identification [4,5]. We propose a two-layer feed forward neural network to approximate nonlinear autoregressive network with exogenous inputs (NARX) for learning "the short and long term dependencies among heartbeat messages over internet", predicting one-step-ahead heartbeat message delay. The inputs are two moving window observations of past heartbeat delays and heartbeat sending time, the output is next heartbeat delay, the output is the one-step-ahead future message delay. The network is trained by back-propagation algorithm, its weights, basis are adjusted by approximate steepest descent rule [6], then adopt current weights and biases to forecast next heartbeat delay.

The remainder of this paper is organized as follow. Section 2 introduces fundamental concept about system model, basic heartbeat strategy and adaptability requirement. Section 3 gives exits approaches. Section 4 describes our proposed approach. Section 5 gives experiment environments and results. Section 6 presents concluding and future work..

2. Basic Failure Detection Algorithm

Our basic failure detection algorithm [2] is generally based on heartbeat message timeouts (Figure.1). It works as the follow: the monitored process p periodically (every η interval) sends heartbeat messages $m_1, m_2, ..., m_l, ...$ to failure detector. The

Project 60772111 supported by National Natural Science Foundation of China
Project 60811140347 supported by NSFC-KOSEF
This research is also supported by China National Key Foundation Research 973 project (No.2003CB314806)

978-0-7695-3507-4/08 $25.00 © 2008 IEEE
DOI 10.1109/CSIE.2009.14

failure detector uses fixed time points $\tau_1, \tau_2, \ldots \tau_l \ldots$ ($\tau_l = EA_l + \alpha$, $EA_l = \sigma_l + D_l$) and the time it receives heartbeat message to determine whether p is trusted or suspected. Anytime, $t \in [\tau_l, \tau_{l+1})$, failure detector trusts p at time t if and only if failure detector has received heartbeat m_l or higher from this monitored process p . Here EA_l denotes the expected arrival time of the l^{th} heartbeat message, α denotes the constant safety margins, σ_l denotes the send time of the i^{th} heartbeat, and y_l denotes the delay of the l^{th} heartbeat.

The accuracy of failure detector depends on predictive techniques of heartbeat message arrival time [2]. According to $EA_l = \sigma_l + D_l$, in synchronized system σ_l is fixed value, the heartbeat arrival time EA_l depends on heartbeat delay D_l . But heartbeat message's transmission over Internet is characterized by high delay and high loss probability as well as arbitrarily variance. The process itself also suffers with slow response. Thus prediction algorithm of failure detector must be sensitive to network and system conditions change, conFigureure and adapt its behavior to avoid incorrect suspicious. Since the real heartbeat delay over internet behaves time-varying and non-linear characters, we can use the learning ability of neural network to approximate linear and nonlinear relationships among successive observations and predict next heartbeat delay.

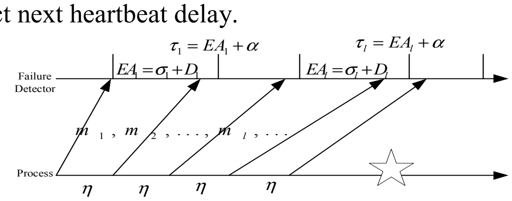

Figure 1. Failure detection based on timeout

3. Existed Predicting Approaches

The observations of heartbeat arrival time can be treated as a time series and time series analysis methods [7] can be used to model and predict next heartbeat with its past several successive observations history. Several traditional approaches can be described by the follow expressions

- **LAST**: the predicted value is the same as the last observation: $\hat{y}(t) = y(t-1)$.
- **MEAN or Moving Averages (MA)**: the predicted value is the arithmetic average of N previous measurements: $\hat{y}(t) = \sum_{i=t-N}^{t-1} y(i)/N$.
- **MEDIAN**: the predicted values is median value of past observation.
- **AR**: auto-regressive (AR) model predicts the next heartbeat $y(t+1)$ as: $\hat{y}(t) = \phi_1 y(t-1) + \phi_2 y(t-2) \ldots + \phi_{na} y(t-na) + e(t)$ where $y(i), i \in \{t, t-1, \ldots, t-na\}$ is heartbeat delay at instant t ., ϕ are coefficients, na is number of coefficients, $e(i), i \in \{t, t-1, \ldots, t-nc\}$ is immeasurable disturbance (i.e., noise).
- **ARX:** Auto-Regressive exogenous (ARX) models heartbeat delay as a SISO (Single-Input and Single-Output) system, the input is heartbeat sending time and output is heartbeat delay.

4. Predicting using NARX

NARX model is well suited for modeling nonlinear systems, It has been shown that gradient decent learning is more effective in NARX networks than in other recurrent neural network architecture that "hidden states" on problems such as nonlinear system identification. Typically NARX network converges much faster and generalizes better than other networks. The defining equation of NARX as: $y(t) = f[u(t-1), u(t-2), \ldots, u(t-n_u), y(t-1), y(t-2), \ldots, y(t-n_y)]$ where $u(t)$ and $y(t)$ represents input and output of system at time t , n_u, n_y are the input and output order, and the function f is a nonlinear function. If we denotes heartbeat delay as output, heartbeat sending time and past heartbeat delays as input, the next heartbeat delay can be regressed on previous values of delay and previous values of sending time (exogenous). NARX (Figure .2) can be used as a predictor, to predict next heartbeat delay. Because the true output is available during the training of the network, instead of feeding back the estimated output, we can use true output. This has two advantages. The first is that the input to the feed forward network is more accurate, the second is that the resulting network has a purely feed forward architecture, and static backpropagation can be used for training. The first input vector u is composed of a moving widow of fixed length time series which is the past successive heartbeat sending time $\{ u(t-1), u(t-2), \ldots, u(t-n_u) \}$ where $u(i)$ is the sending time at time i. The second input vector y is composed of a moving widow of fixed length time series which is the past successive observations $\{\ y(t-1), y(t-2), ..., y(t-n_y)\ \}$ where $y(i)$ is the observation heartbeat delay at time i. The output scalar a_l^2 is prediction value of $y(t)$. Each elements of input vector $u(j), y(k)$ is connected to each neuron (i,l) of hidden-layers input through weights w_{ij}^{11}, w_{lk}^{12}. The number of neurons is equal to the dimension of input vectors, the hidden-layers include n_u, neurons. The i,l th hidden-layer neuron has a summer that gathers its weighted inputs $w_{ij}^{11}u(j), w_{lk}^{12}y(k),$ and bias b_i^1 to form its net input scalar n_i^1, the transfer function f_i^1 outputs a_i^1. Each $a_i^1, i = 1..N$ taken together form N -element input vector which is connect to the neuron of output-layer through weight w_{1i}^{21}, the output-layer only includes 1 neurons, this output-layer neuron has a summer that gathers it weighted input $w_{1i}^{21}a_i^1$ and bias b_1^2 to form its net input scalar n_1^2, the transfer function f_1^2 outputs a^2 which is one-step-ahead prediction of $y(t)$ (we denote as $\hat{y}(t)$).

The forecasting algorithm contains two phases. The back-propagation [8] training phase is used to compute the optimal weights and biases of neural network to minimize prediction error. The forecasting phase adopts current weights and biases to forecast next heartbeat delay. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Basically, training is the process of determining weights and biases of a neural network. The knowledge learned by a network is stored in the weights and biases. It is through weights and biases that neural network can carry out complex linear and nonlinear relation from its input to its output. Our approach employ a supervised learning scheme where the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. As each input vectors is applied to the network, the network output is compared to the target and the training algorithm is used to find the optimal weights and biases that minimize prediction error. In order to minimize the mean square error backpropagation learning is used to update the network weights and biases in the direction where the performance function decreases most rapidly, i.e., the steepest descent algorithm. One iteration of algorithm can be written as: $x(k+1) = x(k) - a(k)g(k)$, where $x(k)$ is a vector of current weights and biases, $g(k)$ is the current gradient, and $a(k)$ is the learning rate.

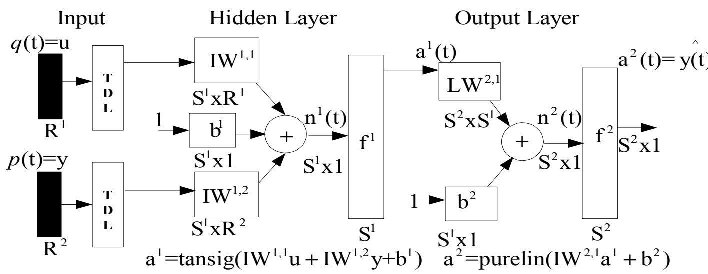

Figure 2. Two-layer feed forward neural network (NARX)

5. Simulation

NS-2 [9] provides functionality to simulate asynchronous communication channels, TCP, UDP, routing, and multicast protocols over wired and wireless networks. We use the following simulated objects to implement our simulation and evaluate our algorithm performance. **Nodes** represent the physical machines. Each Node gets assigned a unique address by ns-2 at creation, and maintains a series of ports that serve as an interface to the network. **Links** correspond to the physical connections among Nodes. Our experiment use duplex, or bi-directional, links, each specified by two endpoint Nodes, link bandwidth, link delay, and queue type. **Agents and Sinks** correspond to monitor process and failure detector. An Agent and Sink pair is associated with a pair of Nodes which define a link. The Agent runs on a port on the source Node, and the Sink runs on a port on the destination Node. Our experiment use UDP Agent and Sink. **Application** represents failure detection service. In our experiment, we use constant Bit Rate (CBR) application to simulate heartbeat messages and use ftp application as background traffics. All predictive methods were implemented using MATLAB. the learning rate, windows rate is 0.1, and momentum is 0.1.

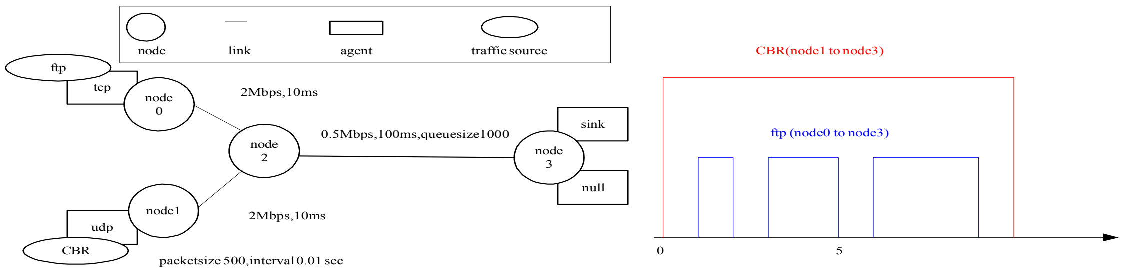

Figure 3. Simulation topology and test scenario with bottleneck link

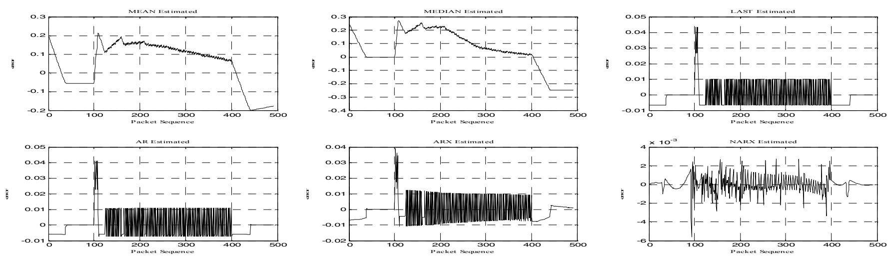

Figure 4. Comparison of prediction result with bottleneck link

Table 1. Comparison of numerical MSE with bottleneck link

Model	LAST	MEAN(20)	MEDIAN(20	AR(2)	ARX(2,2,1)	Neural Net
MSE	5.3151e-005s	9.7810e-004s	0.0012s	6.7777e-005s	5.2970e-005s	7.1763e-007s

- **Bottleneck Link**

The first experiment (experiment 1) is done to compare prediction goodness of fit under dynamic competing traffics. A network topology and test scenario shown in Figure.3 is used in this experiment. Node2 connect node0 and node1 are connected to Node2 with small delay and high-throughput link (2Mbps, 100ms, drop tail queue); Node2 is connected to Node3 with large delay small-throughput bottleneck link (0.5Mbps,100ms, drop tail queue) . From 0 seconds to 10seconds, every 0.01 seconds, monitored process (node1) sends a heartbeat message to failure detector (node3). At 1.0s-2.0s, 3.0s-5.0s, 6.0s-9.0s, node 1 periodically starts and stops ftp traffics. Comparison of prediction error is shown in Figure .4. The bottleneck link (node2—node3) makes packets in queue and causes dynamic packet delay. Since last observation is more reactive to network change, $LAST$, $AR(p)$, ARX can adapt network delay changing in time and obtain better prediction. But in $MEAN$, $MEDIAN$, the last observation is the same weight as all previous observations, they cannot catch this change quickly, result in more binger prediction errors. Neural network predictor can adaptively learn nonlinear and linear characters of heartbeat and makes the smallest prediction error. The numerical MSE (Table 1) shows neural network predictor can provide the best goodness of fit.

- **Link down and up**

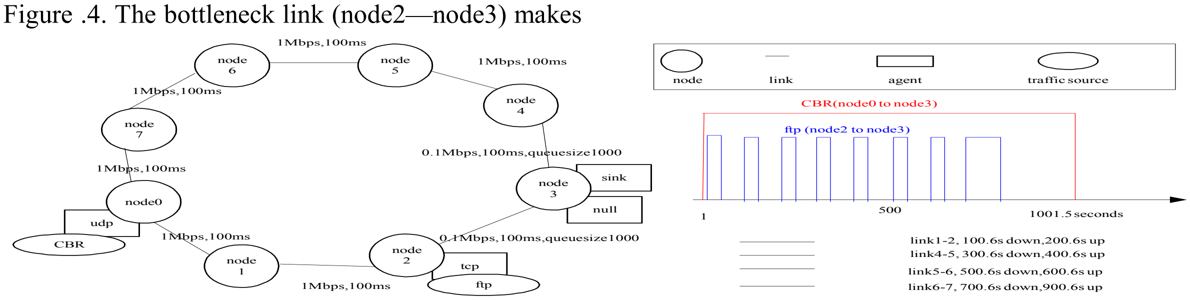

Figure 5. Simulation topology and test scenario with link down and up

The second experiment (Figure.5) is done to investigate prediction goodness of fit under dynamic packet routing path. A network topology and test scenario shown in Figure.8 is used in this experiment. All node are connected as a ring with small delay high-throughput link (1Mbps,100ms, drop tail queue). All nodes employs dynamic shortest routing protocol, from 1 seconds to 1001.5 seconds, every 1 seconds, monitored process (node 0) sends a heartbeat message to failure detector (node3) via routing path 0-1-2-3. At 10s-50s, 110s-150s, 210s-250s, 310s-350s, 410s-450s, 510s-550s, 610s-650s, 710s-800s, node 2 periodically starts and stops ftp traffics. At 100.6s link1-2 down, CBR packet routing path from node 0 to node 3 will change to 0-7-6-5-4-3, At 200.6s link1-2 up, CBR packet routing path from node 0 to node 3 will change back to 0-1-2-3. At 300.6s link4-5 down, at 400.6s link4-5 up, at 500.6s link5-6 down, at 600s link5-6 up, at 700.6s link6-7 down, at 900.6s link6-7 up. Both Figure.6 and Table. 2 show neural network predictor can provide the best prediction result.

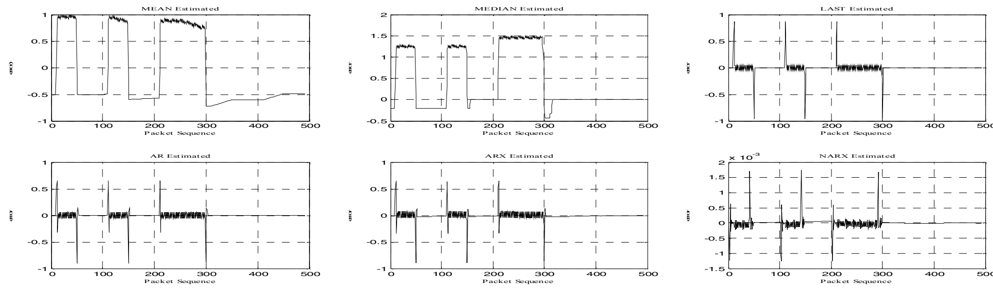

Figure 6. Comparison of prediction results with link down and up

Table 2. Comparison of numerical MSE with Link down and up

Model	LAST	MEAN(20)	MEDIAN(20)	AR(2)	ARX(2,2,1)	Neural Network
MSE	0.0144s	0.1954s	0.2717s	0.0123s	0.0018s	4.1301e-008s

6. Conclusions

Failure detection was a fundamental service in distributed system and its accuracy depends on predicting heartbeat message arrival time. However, Internet dynamic characters bring more tough challenge to it. In this paper, we present a nonlinear autoregressive network with exogenous inputs to learn nonlinear and linear characters of heartbeat messages, perform one-step-ahead prediction.

We hope investigate possible solution under both dynamic delay and loss.

References

[1] Chandra T D, Hadzilacos V, Toueg S. The weakest failure detector for solving consensus. Journal of ACM, 1996, 43(4):685-722.

[2] Chen W, Toueg S, Aguilera MK. On the quality of service of failure detectors. IEEE Trans. on Computers, 2002,51(5):561-580.

[3] Paxson V. End-to-end Internet packet dynamics. IEEE/ACM Transactions on Networking, 7(3):277.292, 1999.

[4] Haykin S, Neural Networks: A Comprehensive Foundation, IEEE Press, New York, NY, USA, 1994.

[5] Waibel A, Hanazawa T, Hinton G, Shikano K, Lang K J, Phoneme recognition using time-delay neural networks, IEEE Trans. Acoustic, Speech Signal Process. 37 (3) (1989) 328–339.

[6] Hagan M T, Demuth H B, and Beale M H, Neural Network Design, Boston, MA: PWS Publishing, 1996.

[7] Pourahmadi M. Foundations of Time Series Analysis and Prediction Theory. New York. John Wiley & Sons, Inc., 2001.

[8] Rumelhart D E, Hinton G E and Williams R J. Learning representations of back-propagation errors. Nature 323, 1986, pp.533-536.

[9] Network Simulator 2, http://www.isi.edu/nsnam/ns/