Impact of Utilizing Forecasted Network Traffic for Data Transfers

Ng Kar Hoong*, Poo Kuan Hoong*, Ian K.T. Tan*, Nithiapidary Muthuvelu*, Lee Ching Seng**
*Systems Innovation Group, Faculty of Information Technology, Multimedia University, Cyberjaya 63100, Selangor, Malaysia
**Penang Design Center, Intel Microelectronics Sdn Bhd, Bayan Lepas FTZ, 11900, Pulau Pinang, Malaysia
ng.kar.hoong@mmu.edu.my, khpoo@mmu.edu.my, ian@mmu.edu.my, nithiapidary@mmu.edu.my, ching.seng.lee@intel.com

***Abstract*—Sharing of information leads to the need to transfer data between geographically distant locations. Identifying the most appropriate time period to execute the data transfer is essential to achieve the best data transfer throughput; e.g. one can forecast network traffic, identify future low network traffic activities between two entities, and plan the data transfer accordingly. This forecasting can be done using Autoregressive Moving Average (ARMA) time series model. In this paper, we conduct an empirical study in a controlled laboratory environment to realise the impact of performing data transfer during the forecasted low network traffic activities. The network information is captured using a network analyzer and post-processed to create stationary data. This data is then passed to ARMA and the forecasting results produced by ARMA is post-processed to derive the forecasted network traffic activities. Comparison is made between the throughputs of the data transfers initiated when the forecasted network traffic is low and when the forecasted network traffic is high.**

***Keywords*— network forecasting, time series, ARMA, data transfer, performance**

I. INTRODUCTION

Network traffic forecasting is an area where statistical methods have been applied with some success. Of the various statistical methods used, the Auto-Regressive Integrated Moving Average (ARIMA) [1][2] time series model has been comprehensively researched with immense success [3][4][5]. The results of the researches have shown that using the ARIMA model, they were able to forecast network traffic with very minimal error rates.

Forecasting of network traffic data has many applications, ranging from intrusion detection [6] to determining the best period to initiate a file transfer [7]. Yaacob et. al. [6] used the ARIMA time series model to forecast future network traffic which it assumes is accurate within certain threshold. This forecasted network traffic is then used to compare with the actual network traffic and since the accuracy threshold was pre-determined, variations greater than the threshold will raise an alert on a possible intrusion. Tan et. al. [7] suggested the use of forecasted network traffic to determine the best period to initiate a small software patch transfer in order to minimize the impact of downloading this patch on other network activities, especially real-time activities such as gaming. They used the Auto-Regressive Moving Average (ARMA) time series model to forecast low network traffic activities. They have shown that the particular model is an efficient computational model which is suitable for short range forecasting in order to initiate small sized software patch downloads. However, the paper does not initiate any form of data transfer but merely provided an indication that the ARMA time series model provides suitable forecasting for the network traffic on a single broadband line.

In this paper we will use the same statistical time series model as described by Tan et. al. [7] to forecast network traffic to study the impact of actual initiation of file transfers when the network traffic is forecasted to be low. When a low network traffic activity is forecasted, it may reflects two possibilities: (i) low network usage during the specified period; (ii) changes in the network bandwidth; e.g. the network administrator may limit the network bandwidth for certain routes.

If it is the former case (i), then the forecasted information can be used to initiate and execute data transfers. However, if it is the latter case (ii), then initiating data transfers during the forecasted low network traffic is counter-productive as it will lead to network congestion, leading to a detrimental data transfer throughput performance. Our studies show that using the information derived from forecasted network traffic, it can be beneficial to perform data transfers during the forecasted low network traffic activities.

In the next section, we describe our method of study. Section III gives an overview of our implementation and testing environment. Section IV provides the results and analysis of our study. Finally we conclude out work in Section V.

II. TOOLS AND METHODOLOGY

In our methodology, we use the following tools; Cronos for the ARMA time series model application, Wireshark for the network analyser and to capture the network packets, and FTP to initiate the file transfer.

A. Cronos and ARMA

Cronos [8] is an open source complete time series analysis package. It provides a number of tools that allow data manipulation and supports a range of forecasting models including the ARMA model used in this paper.

The ARMA model can be explained as a combination of two parts; the Autoregressive (AR) which reflects the historical values; and the Moving Average (MA) which is the

error component [1][2][6][7]. The AR (p) function is represented as:

$$X_t = \sum_{i=1}^{p} \theta_i X_{t-1} + \varepsilon_t \quad (1)$$

where, $t \in \{0,1,2,\ldots\}$, $\theta_1 \ldots \theta_p$ are the parameters of the model that need to be determined, and we use ε_t to represent the constant and white noise error value.

The MA component is the sum of the historical white noise values with the addition of the expected time series value. This MA (q) function can be represented as:

$$X_t = \mu + \sum_{i=1}^{q} \varphi_i \varepsilon_{t-i} + \varepsilon_t \quad (2)$$

where, $t \in \{0,1,2,\ldots\}$, $\varphi_1 \ldots \varphi_q$ are the parameters of the model that need to be determined, μ is the expected value of X_t , and ε_t represents white noise error value.

By combining both (1) and (2), with the assumption of no errors, ARMA can be represented as one equation:

$$X_t = \sum_{i=1}^{p} \theta_i X_{t-1} + \sum_{i=1}^{q} \varphi_i \varepsilon_{t-i} + \varepsilon_t \quad (3)$$

where, $t \in \{0,1,2,\ldots\}$, $\theta_1 \ldots \theta_p$ are the parameters for the AR model, $\varphi_1 \ldots \varphi_q$ are the parameters for the MA model, ε_t represents white noise error value, p is the AR term and, q is the MA term.

This forms a model called ARMA (p,q) of order (p,q) [6]. ARMA should be applied to stationary data. In order to produce high-quality predictive results, an adequate set of historical data will be required by the model for observation.

B. Wireshark

Wireshark [9] is an open source network protocol analyzer that runs on a variety of operating systems. It can provide output in a variety of formats, which serve our data collection well. It captures the network packets and display the content of the packet data. Wireshark is useful for various fields related to networking analysis such as troubleshooting network problems, examining security problems, debugging protocol implementations, learning network protocol, etc.

In our experiment, we collect three hour of network data using Wireshark. The first two hours are used as training data for ARMA and the remaining one hour acts as the testing data.

C. File Transfer Protocol (FTP)

FTP [10] is a standard network protocol used to copy files from one host to another over a TCP/IP-based network, such as the Internet. FTP is built on client-server architecture and it utilizes separate control and data connections between the client and server. For our experiments, FTP is used to send and receive files over the network.

D. Methodology

Figure 1 shows the process flow to forecast the network performance prior to the file transmission. In the first stage, ***Collect Network Traffic Data***, Wireshark is used to capture the number of packets in the network at regular intervals. This information represents the network condition. In the second stage, the number of packets (NP) at each interval is normalized into a log return value by Cronos using the formula given in (4). A log return value reflects the relationship between two consecutive network data.

$$R(t) = \ln(NP_t) - \ln(NP_{t-1}) \quad (4)$$

where,
NP_t = number of packets capture at time t.
$R(t)$ = log return value at time t .

In the third stage, all the log return values are positioned into the ARMA model which will create the appropriate network traffic pattern. This pattern, together with the log return values, is used to forecast the number of packets during the next five steps.

There are two types of forecasting methods: (i) forecasting five steps ahead consecutively; and (ii) forecasting with shifting window (SW) by changing the windows frame of the two hours training data.

Figure 2 explains the terms step and step size used in out experiments. In the second method, the shifting is done after forecasting one step ahead. The forecasted values are then appended back to the last point of the SW before forecasting the next step's value. The windows size remains the same and the method is repeated four times to forecast the four remaining steps. This method is also practiced by Tan et al. [7] in their research.

When using FTP for data transfers, the FTP related packets are also included in the transfers. Hence, in order to remain the integrity of our experimental results, we removed the FTP packets from our one hour testing data using the filters available in Wireshark. The remaining number of packets project the actual network information. We compare the forecasted result with the remaining number of packets to compare the accuracy of the forecast.

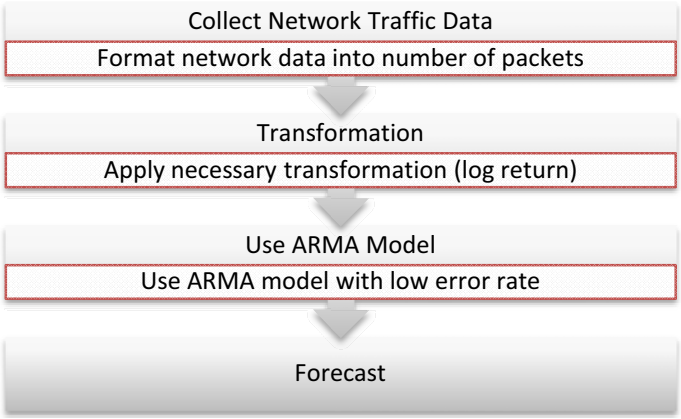

Figure 1. Process flow for forecasting network pattern.

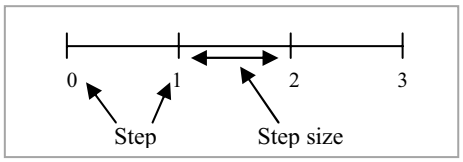

Figure 2. Step and step size.

III. EXPERIMENTAL SETUP

In this section, we conduct three phases of experiments.

- Phase I: Determine the ARMA p and q orders with the assistance of the log likelihood function [11] that is provided in the Cronos application. The *log likelihood* function is a statistical method used to estimate the parameters of time series models.
- Phase II: Using the ARMA p and q orders determined, we proceed to initiate file transfer for a 10 Megabytes (MB) file.
- Phase III: Repetition of Phase II using 1 Gigabytes (GB) file.

A. Phase I: Determining ARMA p and q Order

The experimental setup for Phase I is shown in Figure 3. We conduct the experiments in a controlled environment where machines are located within the campus domain, which is a Megabits network infrastructure. Wireshark is installed on both Node 1 and Node 2 in order to capture the number of packets for the duration of two hours of training period.

We capture two hours of network traffic data which is then divided into various step sizes. This is used as the inputs to Cronos, where the log return transformation is done. The Cronos then cycles through all ARMA orders for p = 0 to 10, and q = 0 to 10 (iterative increment), and checks the *log likelihood*. The smallest absolute value of the *log likelihood* function is then selected as the best p and q order [11] to be used for our ARMA model of certain step size.

B. Phase II: 10 MB File Transfer

Using the ARMA (6,4) model with two hours of training data (historical data) in 30 seconds step size, we run the experiment to transfer a 10 MB file and observe the throughput performance for the first few steps. In order to simulate a real environment, we configured our environment as shown in Figure 4. Instead of conducting file transfers within the high-speed campus network, we initiated file transfers between a node that is connected to the local Internet Service Provider via ADSL broadband link (1.2M bandwidth) and a node within our campus (more than 32M bandwidth).

A detailed explanation of the procedure used is provided in Figure 5.

C. Phase III: 1 GB File Transfer

As an endorsement of the previous phase, a 1 GB file transfer is also conducted to verify that the results obtained is suitable for large file transfers.

IV. RESULTS AND ANALYSIS

A. Analysis - Phase I: Determining ARMA p and q Order

Phase I's objective is to find the most suitable order of ARMA model and the relevant step size. The model selection is done by iteratively incrementing the order of the model for each step size to find the suitable p and q parameters. These suitable p and q parameters should result in the maximum *log likelihood* for that particular step size.

As for this purpose, we use step sizes of 30 seconds, 60 seconds, 90 seconds, and 120 seconds. We selected multiple step sizes as different step sizes affect the pattern of the two hours training data in different ways. The impacts of various step sizes are reflected on the forecasted results.

From our experiments, we select the most suitable ARMA order for each step size based on maximum *log likelihood* . These models are shown in Table 1. From these results, we select the best model by using the the Mean Squared Error (MSE) as in the equation (5).

$$MSE(x) = \frac{1}{n}\sum(Actual_i - Predicted_i)^2 \quad (5)$$

where,
x = step 1, step 2,...., step 5
$i = i^{th}$ observation for step x
n = number of observations for step x

The accuracy of one step ahead forecast is measured by using the MSE. The average MSE is then calculated for all the five steps. Table 1 also shows the average MSE for each step size. Out of these four step sizes, the ARMA order model with the smallest average MSE is chosen as the most accurate combination for both ARMA order model and the step size. Since the forecasted step 1 is important in phase II, the MSE values for step 1 are recorded in Table 1 as well.

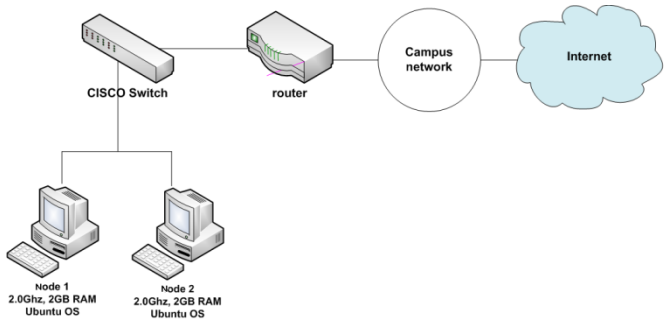

Figure 3. Experimental setup for experiment Phase I.

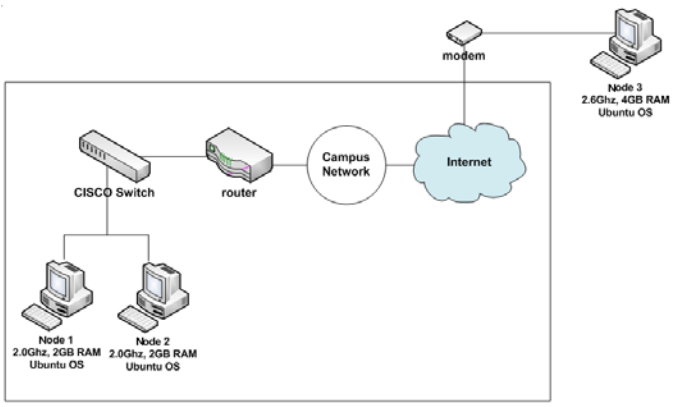

Figure 4. Experimental setup for file transfer.

1. The Wireshark in both Node 1 and Node 3 will collect the network data for the duration of two hours.
2. After the two hours,
   - 2.1 Cronos will forecast the number of packets for the next five step size using the two hours network data collected previously by Wireshark. The forecasted number of packets will be compared with the actual network performance at the end of the experiments.
   - 2.2 Initiate the transfer of 10 MB file from Node 1 to Node 3 during forecasted low number of packets.
   - 2.3 The Wireshark will keep capturing the network data throughout the experiments.
   - 2.4 Wait till the 10 MB file is successfully transferred.
   - 2.5 Collect the network data captured by Wireshark in the last two hours.
   - 2.6 Continue with step 2.1.
3. Compare the forecasted network data with the actual data for further

Figure 5. Procedure for file transfer.

During the training period, we notice that the Wireshark can capture a maximum of 30,000 packets at a particular time. From the results shown in Table 1, it is noted that the MSE values are within the acceptable range as compared to the actual network traffic data which can reach up to 30,000 packets. The difference between the forecasted and the actual network traffic packets is relatively small.

The MSE values in Table 1 show that ARMA (6,4) with step size of 30 seconds indicates the minimum average MSE and MSE for step 1.

TABLE 1. AVERAGE MEAN SQUARED ERROR FOR EXPERIMENTAL PHASE I

Step Size	Model Order	Average MSE	MSE Step 1
30	ARMA (6,4)	5207	1676
60	ARMA (3,1)	5951	3314
90	ARMA (10,6)	6926	4162
120	ARMA (6,4)	9115	3721

B. Analysis - Phase II: 10 MB File Transfer

In this paper, we assume that when the forecasted number of packets shows relatively low value, it represents a low network traffic at that forecasted time. In this emphical study, we test our forecasting ability in two situations: (i) we inititae the transfer of a 10 MB file at the particular time when the forecasted network traffic is low (Figure 6); and (ii) we inititae the transfer of a 10 MB file at the particular time when the forecasted network traffic is high (Figure 7).

The experiments of these two situations are conducted at different time period. Hence, it can be noticed that the number of packets indicated at Y-axis of both the figures range with large difference.

Table 2 consists of a sampling of 7 runs indicating the number of packets at step 0, actual packets at step 1, and the forecasted number packets for step 1 together with the overall completion time for the transfer of 10 MB file. The file transfers are executed at step 1.

Figure 6 is when the forecasted traffic is low with corresponding actual low network traffic whilst Figure 7 is when the forecasted traffic is high with corresponding actual high network traffic. The time needed for completing the file transfer is 518 seconds and 610 seconds respectively. This indicates that initiating a file transfer when the next step of the forecast is low will provide a better file transfer throughput.

During the file transfer, Figure 6 successfully sends over 30% of the file at step 5, while Figure 7's progress is 28%. From this result, it can be deduced that the low network traffic forecasted and actual as in Figure 7 is not due to limitations of the network bandwidth. This can be concluded as during the transfer of the files, more network data packets are observed over the same time period.

TABLE 2. FORECAST RESULTS AND FILE TRANSFER TIME

#	Step 0	Step 1 (Actual)	Step 1 (Forecast)	Transfer Time (sec)
1	14566	14594	12090	518
2	34041	16856	17869	532
3	3516	1279	6691	515
4	1744	2567	1744	517
5	9563	14453	9093	519
6	1697	3188	1954	610
7	2242	5239	2316	629

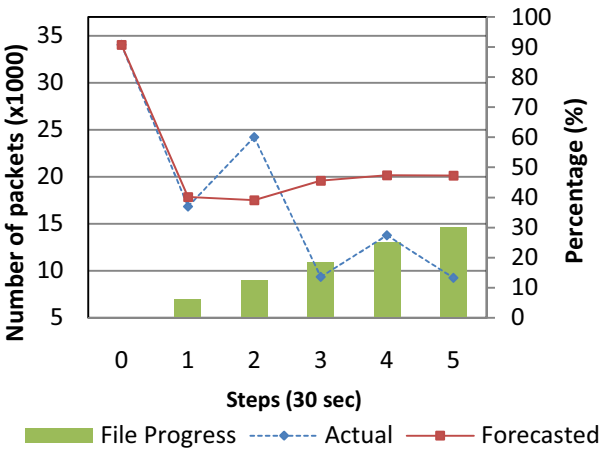

Figure 6. Forecasted low network traffic using ARMA(6,4) with step size of 30 seconds and the 10 MB file transfer performance.

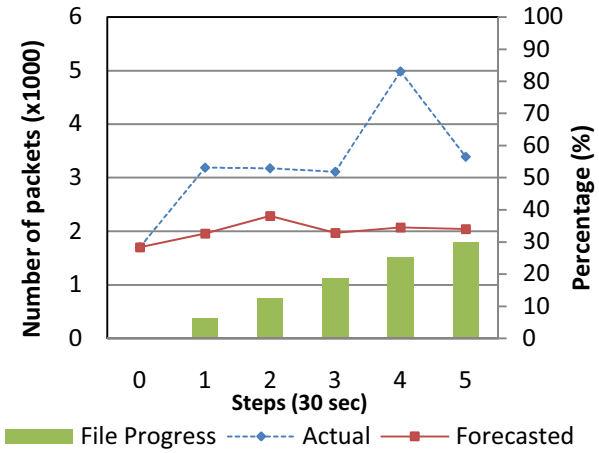

Figure 7. Forecasted high network traffic using ARMA(6,4) with step size of 30 seconds and the 10 MB file transfer performance.

Time series forecasting accuracy is inversely proportional to the number of steps forecasted. For example, Figure 6 shows a slight forecast inaccuracy from step 3 onwards where one can notice a large difference between the forecasted and the actual number of packets. This is due to the time series forecasting method which depends solely on the historical data to determine the future points and if we forecast more points ahead, the accuracy will decrease.

C. Analysis - Phase III: 1 GB File Transfer

In Phase III, we transfer large files of 1 GB over the network. Figures 8 and 9 show the forecasted five steps ahead for the network traffic. The time taken to transfer 1 GB file over the network is 63718 seconds and 65068 seconds respectively.

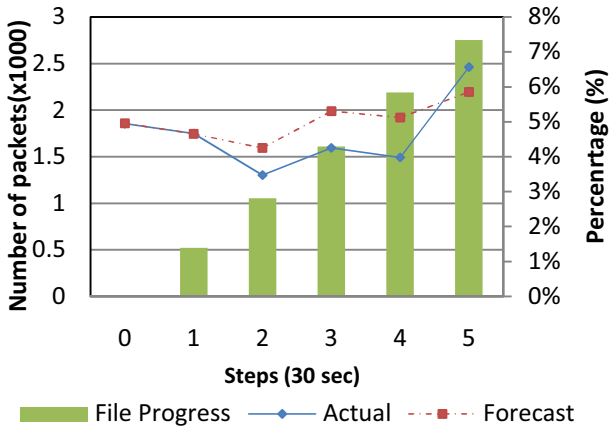

Figure 8. Forecasted low network traffic using ARMA(6,4) with step size of 30 seconds and the 1GB file transfer performance.

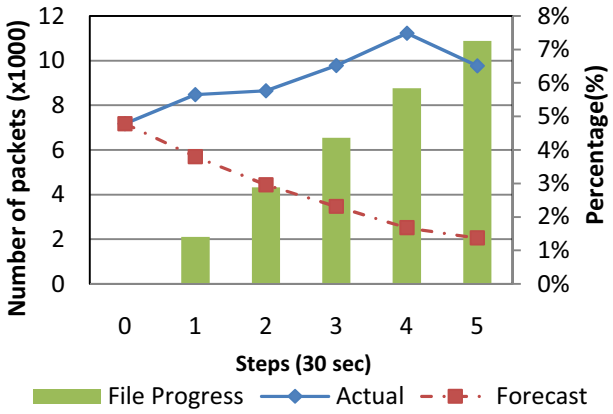

Figure 9. Forecasted low network traffic using ARMA (6,4) with step size of 30 seconds and the 1GB file transfer performance.

It shows that after 30 seconds, the amount of data transferred in Figure 9 is approximately 30% less than Figure 8. However, as shown in both the figures, after 5 steps the difference of data transfer is insignificant. This shows that in order to transfer a large file size, forecasting alone is insufficient and there is a need to combine other technique such as file chunking.

I. CONCLUSION

In this paper we forecast the network traffic with different windows sizes and step sizes. From our experiments, we determined that the most appropriate model to be used is the ARMA (6,4) with step size of 30 seconds. This model is capable of forecasting for short range network traffic.

In our studies, we have also shown that forecasting network traffic can be used to enable more efficient large file transfers and the forecasting using the ARMA time series model shows great promise of being able to be included as an intelligent model for a high throughput performance file transfer application.

The authors note that a technique to divide the files into smaller sizes and transferring them when low network traffic is forecasted would lead towards a better efficient use of network bandwidth.

ACKNOWLEDGMENT

We would like to thank Intel Penang Design Centre for their research funding.

REFERENCES

[1] Brockwell, Peter J. and Davis, Richard A., "Introduction to Time Series and Forecasting (2nd Edition)", Springer-Verlag, 2002, pp. 151-192.

[2] Box, George E. P. and and Gwilym M. Jenkins, "Time Series Analysis: Forecasting and Control, revised ed.", Holden-Day, Oakland, 1976.

[3] Jun Lv; Tong Li; Xing Li; , "Network Traffic Prediction Algorithm and its Practical Application in real network," NPC Workshops, IFIP International Conference on Network and Parallel Computing , Sept. 2007, pp.512-517.

[4] Sangjoon Jung, Chonggun Kim, Younky Chung, "A Prediction Method of Network Traffic Using Time Series Models", Proc. Computational Science and Its Applications (ICCSA), 2006, pp.234 – 243.

[5] B. Zhou, D. He, and Z. Sun. "Network traffic modeling and prediction with ARIMA/GARCH", Proc. 3rd International Working Conference: Performance Modelling and Evaluation of Heterogenous Networks (HET-NETs'05), 2005.

[6] Asrul H. Yaacob, Ian K. T. Tan, S. F. Chien, H. K. Tan, "ARIMA Based Network Anomaly Detection", Proc. *2nd International Conference on Communication Software and Networks (ICCSN 2010)*, Singapore, 2010 pp. 205 – 209

[7] Ian K.T. Tan, Poo Kuan Hoong, Chee Yik Keong, "Towards Forecasting Low Network Traffic for Software Patch Downloads: An ARMA model forecast using CRONOS", Proc. *International Conference on Computer and Network Technology (ICCNT 2010)*, Bangkok, Thailand, 2010, pp. 88 – 92.

[8] Cronos, http://www.codeplex.com/cronos/. [Accessed 19 December 2009.

[9] Wireshark, http://www.wireshark.org/. [Accessed 17 July 2010]

[10] J. Postel and J. Reynolds. "RFC 959: File transfer protocol (FTP). Technical report", University of Southern California Information Sciences Institute, October 1985.

[11] Sowell, Fallaw, "Maximum likelihood estimation of stationary univariate fractionally integrated time series models", Journal of Econometrics , July-September 1992, pp. 168-188