IJCSN International Journal of Computer Science and Network, Volume 3, Issue 2, April 2014
ISSN (Online) : 2277-5420
www.IJCSN.org

A Best Fit Mathematical Model for Maximizing GridFTP Throughput Based on Real and Simulated Network Conditions

[1] **Ioannis Barbounakis,** [2] **Pantelis Askordalakis,** [3] **Michael Zervakis,** [4] **Demosthenes Akoumianakis**

[1, 2] Department of Electronics Engineering, Technological Educational Institute of Crete, Chania, Crete, Greece.

[3] Department of Electronic & Computer Engineering, Technical University of Crete, Chania, Crete, Greece.

[4] Department of Information Engineering,Technological Educational Institute of Crete, Herakleion, Crete, Greece.

Abstract - Grid Computing refers mainly to resources which are geographically distributed over a wide-area network (WAN). Therefore, the need for a high throughput transfer protocol is essential. This has led to the development and adoption of the GridFTP protocol. Although, several attempts have been made to achieve higher throughputs, they appear not to be optimal. In this work, we present a different approach to the problem. First, the throughput is modeled in terms of the following parameters: File-size, Network Bandwidth, Round Trip Time (RTT), number of parallel streams and TCP sender socket buffer size. Second, three different models are adopted: 1) Multiple Linear Regression, 2) Multiple Non-Linear Regression and 3) Linear Minimum Mean Square Error (LMMSE). Finally, a theoretical analysis is performed, assisted by a set of simulations and real-world experiments, which yields the best-fitting model.

Keywords - *GridFTP, TCP parallel streams, TCP socket buffer size.*

1. Introduction

During the last years, the rapid development in the area of Grid Computing [1] has increased considerably the need for large volume data transfers among distributed computing resources. At the same time, it has also been made clear that the current file transfer protocols are unable to cope with these ever-increasing demands, if they keep operating in a static rather than in a dynamic manner. Under these new conditions, the grid research community has identified the underlying TCP limitations as the main causes that prohibit the current FTP protocol from being dynamically adjusted to the ever-changing network conditions. These limitations are: 1) Inability to support multiple-stream, parallel transfers, since it has been developed as a one-stream transfer protocol. 2) TCP uses a buffer at both ends whose size is configured by the operating system in a static manner. This value is suitable for low-scale networks (LANs). 3) TCP block-size boundaries are not set dynamically by the kernel, but instead they could be set by each application. However, very few applications are capable of setting such parameters.

Hence, an improved FTP version known as GridFTP [1] has been the outcome of research in the field to overcome these TCP limitations. Although, it integrates some additional features such as control of TCP sender socket buffer size, number of parallel TCP streams and block-size, it does not provide a complete solution to the problem “what should be the optimal values for these parameters?”. Moreover, by having current operating systems such as Microsoft Windows & Linux to pre-set parameters such as the TCP sender socket buffer size on behalf of the applications is not optimal under any network condition.

Various efforts towards the estimation of these optimal GridFTP parameters as well as methods of GridFTP auto-configuration are presented in [2] - [4].

In paper [2], an extension of the GridFTP protocol is proposed, which determines the required TCP sender socket buffer size according to the Bandwidth Delay Product (BDP). Particularly, it is based on a one-time measurement of the BDP, i.e. Round Trip Time (RTT) and bottleneck (available) bandwidth. Under real network conditions though, the BDP value varies over time. Moreover, other network parameters such as the number of TCP parallel streams and non-network parameters such as file-size are not taken into consideration. Finally, the proposed mechanism allocates twice of the BDP per TCP connection, which is inefficient regarding memory allocation according to our conclusions.

In paper [3], a fluid-flow model has been adopted for the TCP control mechanism when applied to large scale IP networks. Here, multiple TCP connections are modelled as independent continuous-time systems and the bottleneck router is modelled as another single continuous-time system. An analysis of the final model is then performed. As a result, the optimal number of TCP connections and the TCP sender socket buffer size are computed. However, it should be mentioned that the

RTT, the packet loss probability and the bottleneck bandwidth should be known in advance. Moreover, only server-to-client transfers are taken into consideration, whereas traffic on the control channel is neglected. Although, they use the TCP sender socket buffer size equal to BDP in their ns-2 based simulations, they suggest that TCP sender socket buffer size should be larger than the BDP value.

In paper [4], an automatic configuration mechanism, based on the work done in [3], is proposed. It is, however, unable to apply to third-party transfers since the configuration is fully performed at the client side. This way, only client-to-server transfers should benefit, and hence this mechanism should not be integrated in an updated version of the GridFTP protocol. However, it seems that such a mechanism should be more beneficial to Grid transfer protocols, if it was implemented in the middleware.

Contrary to the previous works, our approach to the problem of GridFTP throughput maximization moves to the opposite direction. First, we started our work by performing a set of real-world experiments over wide area network conditions and by recording the measurements. Afterwards, we experimented with the network throughput behaviour, via a network emulator, while varying the following network and non-network parameters: 1) File-size, 2) RTT, 3) Network Bandwidth, 4) Number of parallel streams and 5) TCP sender socket buffer size. Initially, we considered BDP as the only throughput-affecting parameter, which characterizes the current network conditions. However, it was soon discovered that BDP cannot be considered equivalent to the various bottleneck bandwidth and RTT value pairs with equal BDP products, i.e. the pair (100Mbps, 70ms) has the same BDP value as the pair (1Gbps, 7ms). The reason for that has to do with the ability to exploit a greater number of parallel streams under higher bandwidth conditions. A behavioural analysis was followed, to find the best-fitting model to the network throughput estimator. Through this estimator, each application and hence GridFTP would be able to pre-configure (right before the data transfer) the best network parameters (number of TCP parallel streams and TCP sender socket buffer size) in terms of maximum throughput.

Unlike [2], our work focuses not only on multiple TCP streams but also on TCP sender socket buffer size. Whereas, [3] and [4] also focus on these two parameters, their proposed mechanism has only been tested under ns-2 and is only based upon BDP. Moreover, it cannot be applied to third-party transfers.

Section 2, refers to the GridFTP protocol and its basic features. Section 3 introduces the throughput measurement setup that has taken place. The measurements are modelled mathematically and statistically in Section 4. Section 5 analyses the models introduced in 4 with respect to the achieved throughputs. Finally, Section 6 summarizes the main conclusions in this work and proposes new visions.

2. GridFTP

GridFTP [1] has its origins to the well-known standard FTP protocol; it additionally provides secure as well as more efficient data transfers between grid nodes.

The GridFTP additional features include: 1) **G**rid **S**ecurity **I**nfrastructure (GSI) [5] integration, 2) Third-party control of data transfer: the GridFTP separates control and data channels, enabling third-party transfers, that is, the transfer of data between two end hosts, mediated by a third host, 3) Parallel data transfers with multiple TCP streams being used between two hosts as shown in Figure 1, 4) Multi-host striped data transfer, 5) Partial file transfer, 6) Manual control of TCP buffer/window sizes, 7) Support for reliable and restartable data transfer in case of failure.

The GridFTP protocol specifies two modes of data transfer: 1) Stream Mode (Mode S), 2) Extended Block Mode (Mode E). *Stream Mode* is implemented in FTP servers where bytes flow in order, over a single TCP connection. There are no advanced features in this mode. GridFTP defaults to this mode so as to be compatible with normal FTP servers.

On the other hand, *Extended Block Mode* constitutes an extension of the *Stream Mode* where data can be sent over the data channel in blocks. Each block consists of 8-flag bits, a 64-bit integer indicating the offset from the start of the transfer, and a 64 bit-integer indicating the length of the block in bytes followed by a payload of length bytes. Because the offset and length are provided, out-of-order transmission is acceptable, i.e. the 12th block could arrive before the 7th. This also enables parallelism and striping.

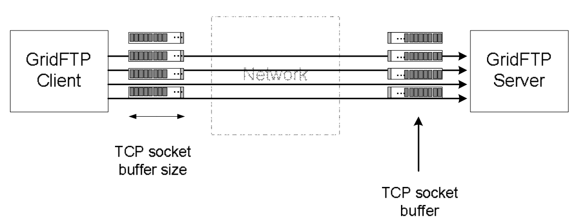

Fig 1: TCP parallel streams

3. Throughput Measurement Setup

In this section, we describe two different network configuration setups to measure the throughput of the GridFTP protocol. In both cases, we commit two hosts running the client and the server applications of the GridFTP protocol. In the first setup, we have made use of the LANforge network emulator [7] as shown in Figure 2, whereas in the second setup, we have exploited the already established Grid infrastructure between Technical University of Crete (TUC) and University of Plymouth (UoP).

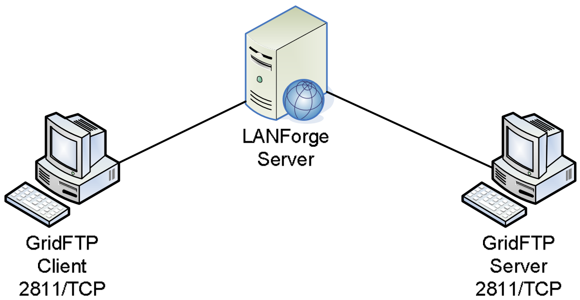

Fig 2: LANForge Topology

LANForge is an advanced network emulator, which provides a handful of network parameters enough to extensively test the GridFTP protocol locally without the inconvenience of having to coordinate different administrative domains.

3.1 Choosing the Parameters for LANForge Setup

Regarding file-size, we have used 100MB, 1GB and 10GB values. RTT values were chosen in the range from 20ms to 100ms because of corresponding value measurements under real network setup. For the same reason as previously, network bandwidth values were selected in the range from 45Mbps to 155Mbps. As for the number of TCP parallel streams, we have worked on the most commonly adopted values (8, 16, 32, and 64). In our set of simulation runs, choosing more than 64 parallel streams caused no remarkable improvement on the throughput. Finally, sender TCP socket buffer size values were chosen in the range from the default kernel value of 128KB up to 1MB. In all cases, we have cautiously selected the (RTT, Bandwidth) pair values so that the equivalent BDP value remains well below the maximum TCP sender socket buffer size value being used in our experiments. Finally, we have inserted a packet-drop frequency of 10^{-3} packets/sec and a maximum jitter value of 4ms at both ends of the WAN link so that LANForge setup resembles more to a real network setup.

3.2 Results Produced from LANForge Setup

In our first set of simulation runs, we logged the throughput of 100MB file-transfers. The throughput reaches a maximum at a specific pair of values for the number of parallel streams and the TCP sender socket buffer size [16 streams, 854KB buffer size] (Fig. 3). For all the other pairs of values, the throughput remains at lower levels. For each case with either 32 or 64 parallel streams, it seems that the extra throughput gain is compensated from the time required to reassemble the packets from these extra streams.

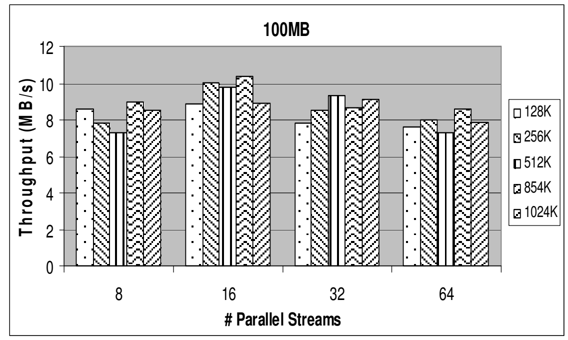

Fig 3: Throughput vs. Parallel streams for a 100MB file-transfer, using different TCP sender socket buffer size values

Figure 4 summarizes the results from the 1GB file-transfers. In this scenario, it seems that the TCP sender socket buffer size has sufficient time to reach its maximum value, which corresponds to the maximum throughput the file transfer can take from the network. That's why, whatever the number of parallel streams and the TCP sender socket buffer size used in the simulations, the throughput reaches the same maximum. This behavior is further overstressed by the 10GB file-transfers.

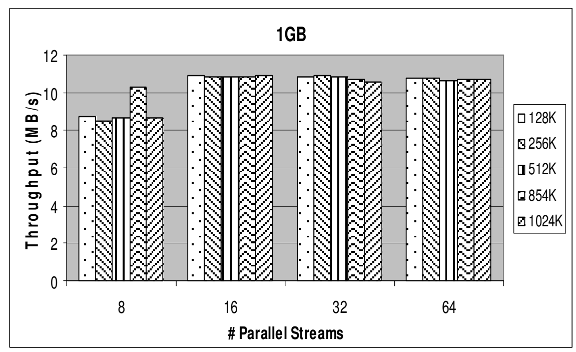

Fig 4: Throughput vs. Parallel streams for a 1GB file-transfer, using different TCP sender socket buffer size values

3.3 TUC-UoP Grid testbed

During the establishment of a small Grid testbed between TUC and UoP, we had the opportunity to test several GridFTP file transfers under real network conditions. These tests were run at different times of the day and for a time period longer than a month in order to cover the whole range of network conditions. We made use of the Ethereal Monitoring Tool [8] to measure both RTT and bottleneck bandwidth. The measured values were 70ms and 50Mbps respectively. They produce a BDP value of 427 KB, more than three times the TCP default sender/receiver socket buffer size of 128KB.

Figure 5 presents the relevant measurements. It's worth noting the proximity between the simulated and the measured throughput values for the case of 100MB file-transfers.

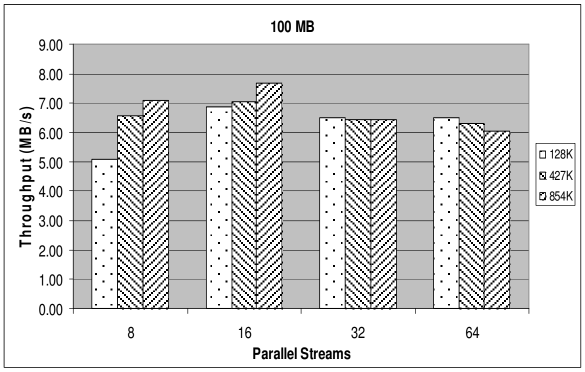

Fig 5: Throughput vs. Parallel streams for a 100MB file-transfer, using different TCP sender socket buffer size values between TUC and UoP

4. Throughput Modeling

As mentioned in the introduction, a fluid-flow model has been adopted by [3]. Applying steady state analysis, the GridFTP throughput is given as a function of the TCP sender socket buffer size "*W*", the number of parallel TCP connections "*N*", the round-trip time "*R*", the bottleneck link bandwidth "*B*" and the packet loss probability "*p*" as shown in Eq. 1:

$$\overline{G}^{*} \approx \min\left(\frac{NW}{R}, \frac{N(1-p)}{2R}\left(-3+\frac{\sqrt{6+21p}}{\sqrt{p}}\right)\right) \quad \textbf{(1)}$$

The optimal number of parallel TCP connections is obtained by determining "*N*" that maximizes Eq. 1.

In our approach however, we have already setup a network testbed as described in the previous section. Through the availability of the necessary network measurements, we are able to observe the throughput behaviour in terms of each parameter.

Our first conclusion was that the throughput depends more or less on the parameters used in the network setup, which are file-size, RTT, Network Bandwidth, number of TCP parallel streams and TCP sender socket buffer size. On the other hand, parameters such as packet drop frequency and jitter did not produce any noticeable throughput decrease, when introduced on the simulations. That led us to exclude those parameters from the mathematical modeling.

Reviewing the relevant literature, we found that the most suitable models that could apply to the observed throughput behaviour would be the Linear Regression model [9]**Error! Reference source not found.**, the Non-linear Regression model [10]**Error! Reference source not found.** and the classical Least Linear Minimum Mean Square Error estimator [11]. Thus, we applied those models to the recorded measurement results and attempted to find the best-fitting model.

In particular, we have properly modelled the GridFTP file transfer throughput as a *function* of the network and non-network parameters mentioned above according to the following formula:

$$Y = f(X, A) + \varepsilon \quad \textbf{(2)}$$

Where, "*Y*" is the throughput vector, "*X*" is the matrix that contains the combination of the five parameters, "*A*" is the parameter weighting vector and "*ε*" is the error vector.

For each applied model, the steps below were followed:

1. Computation of the "*A*" vector.
2. Estimation of the throughput "*Y*" vector based on "*A*" vector from step "1" and the parameter "*X*" matrix.
3. Statistical processing of the deviations between the estimated "*Y*" vector and the observed "*Y*" vector.

The last step is used as an indicator of which model best fits to the throughput maximization problem.

4.1 Multiple Linear Regression

The linear model has the following formula:

$$Y = XA + \varepsilon \quad \textbf{(3)}$$

$$y_i = a_1 x_{i1} + a_2 x_{i2} + a_3 x_{i3} + a_4 x_{i4} + a_5 x_{i5} + e_i$$

where, $i = 1,2,\ldots,n$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}_{(nx1)} \qquad A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}_{(5x1)} \qquad \varepsilon = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix}_{(nx1)}$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ \vdots & \cdots & & & \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} & x_{n5} \end{bmatrix}_{(nx5)}$$

Least squares is used to solve the above problem in order to estimate the parameter vector "*A*" given the observations "*Y*" and "*X*". The *least squares* procedure determines the "*A*" vector that minimizes the sum of squares given by

$$\zeta^2 = \sum_{i=1}^{n} (y_i - a_1 x_{i1} - a_2 x_{i2} - a_3 x_{i3} - a_4 x_{i4} - a_5 x_{i5})^2$$

Using calculus the resulting equation for "*A*" is given by

$$A = (X^T X)^{-1} X^T Y \quad \textbf{(4)}$$

4.2 LMMSE Estimation

In this model "*Y*" and "*X*" are considered as *Random Vectors*. Assuming that these vectors are dependent, the following question, which is described in Eq. 5, comes up: "given the random vectors "*X*" and "*Y*", what is the vector "*A*" that *best* estimates or approaches the random vector "*Y*" ?".

$$\hat{Y} = XA + \varepsilon \quad \textbf{(5)}$$

The corresponding minimum-mean-square-error (m.m.s.e for short) matrix of Eq. 5 is given by

$$m.m.s.e = E(y - \hat{y})(y - \hat{y})^* = R_x - AR_{xy} \quad \textbf{(6)}$$

The optimum choice "*A*" that minimizes Eq. 6 (i.e. minimizes the mean-square error in the estimator of each component of the vector "*Y*") is given by

$$A = R_{X,Y} R_X^{-1} \quad \textbf{(7)}$$

where, $R_{X,Y}$ is the cross-correlation matrix between "*X*" and "*Y*" and R_X^{-1} is the inverse auto-correlation matrix of "*X*".

4.3 Multiple Non-linear Regression

Unlike the Linear Regression (which is a simple straight relationship), the Non-linear Regression is not defined by a specific relationship in the literature. In other words, this type of regression is case dependent. Therefore, by analyzing and graphically representing (see Figures 3 and 4) the throughput measurements in terms of each parameter, we came to the conclusion that the throughput changes exponentially to the parameter variations as shown in the following formula:

$$Y = e^{XA} + \varepsilon \quad \textbf{(8)}$$

After linearization, Eq. 8 becomes:

$$\log Y = XA + \log \varepsilon \quad \textbf{(9)}$$

Note that, Eq. 9 is now a linear equation which can be solved in the same way as with the Linear Regression model.

5. Throughput Measurement Analysis

According to the three models mentioned in section 4, the "*A*" vector has been estimated based on the simulation results using Matlab [12]

From the first view, it seems that the Linear Regression model is the most simplified model of the three because it assumes that the throughput vector "*Y*" changes proportionally to the "*X*" parameter matrix. Of course, the linear model adoption proved not to be always valid. This happens especially during large file transfers, where there is enough time for the throughput to reach its maximum. Under this condition, we do not expect any throughput increase by further increasing the number of parallel streams (see Fig.4).

On the other hand, the LMMSE adopts a linear relationship between the parameters and the throughputs, which are going to be estimated. This is a *stochastic* model and is often called by the name "Linear Least Mean Squares", whereas the previous model is *deterministic* and is called "Linear Least Squares". In this case, the estimated throughput results diverge from the measured values. This happens mostly due to the discrete nature of the parameter sets (i.e. Number of Parallel Streams is limited to powers of 2.) leading to a small degree of variance in the parameter sets that have been chosen.

Finally, the Non-linear model is based on the multidimensional "*X*", "*Y*" data, where "*Y*" is exponentially related to "*X*" as shown in Eq. 8. After linearization, we adopt the Least Squares method to obtain values for the vector "*A*" the same way as with the linear model. In this case, the theoretical results were found to be closer to the measurements than the other models.

In performing throughput measurement analysis, we calculated the deviation between measured and estimated throughput over all possible configuration parameter setups. The results showed that the LMMSE model was far behind the other two models in estimating the throughput. Limiting our interest to the Linear and Non-linear Regression models, we calculated the percentage of the simulation scenarios whose deviation was below some predefined values.

The first deviation value was chosen to be 2.2 MB/s because the Non-Linear model estimations were below this value for the whole of the configuration scenarios (100%). After that, we had to reject the few cases where the Linear Model produced throughput values with deviations larger than the above limit. In an attempt to observe the behavior of the two models, while decreasing the maximum acceptable deviation upper limits, we chose the 1.6MB/s and 1.1MB/s, as shown in Fig. 6. It is obvious, that the Non Linear model outperforms the Linear Model in the first two deviation values. However, when we limit the deviation to 1.1MB/s, both models provide similar estimation capabilities.

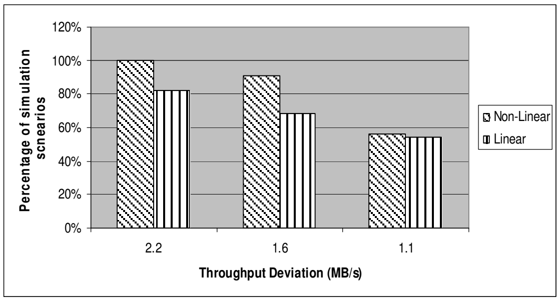

Fig. 6: Throughput Deviation

6. Conclusions

In this work, we have set as a goal to find the best-fitting model among Linear, Non-Linear and Stochastic models, which by pre-selecting the optimal values for the number of TCP parallel streams and TCP sender socket buffer size), would estimate closely the observed GridFTP throughput. Our efforts have been based not only on real-condition but also on emulation results and have utilized both network and non-network parameters.

As explained in section 5, the model, whose estimations best fit the observed throughput measurements, is the Non-Linear (exponential) model. As a next step, we have been working on developing the necessary gridftp library modules, which could automatically adjust GridFTP in order to achieve the maximum available network throughput within any Grid infrastructure. Moreover, we extend our approach in order to cover every recent network technology and capacity.

Acknowledgments

We would like to express our gratefulness to both Mrs. Lingfen Sun and Mr. Pin Hu from the University of Plymouth, United Kingdom, for their assistance in running the GridFTP trials between our universities. Many thanks to the creators of LANForge network emulator for their licensing and their helpful hints. This work has been partially supported by the research project “Electronic Villages for Business Administration and Innovation”, which has been funded by the General Secretariat of Research and Technology in Greece.

References

[1] GridFTP, http://www.globus.org/toolkit/docs/4.0/data/gridftp/

[2] S. Thulasidasan, W. Feng, and M. K. Gardner, “Optimizing GridFTP through dynamic right-sizing”. In Proceedings of IEEE International Symposium on High Performance Distributed Computing, June 2003.

[3] T. Ito, H. Ohsaki, and M. Imase, “On parameter tuning of data transfer protocol GridFTP in wide-area Grid computing”. In Proceedings of Second International Workshop on Networks for Grid Applications (GridNets 2005), pages 415–421, Oct. 2005.

[4] T. Ito, H. Ohsaki, and M. Imase, “Automatic parameter configuration mechanism for data protocol GridFTP”. In Proceedings IEEE Symposium on Applications and the Internet (SAINT’06), 2006.

[5] Grid Security Infrastructure GSI,

[6] http://www.globus.org/security/overview.html

[7] LANForge, http://www.candelatech.com/

[8] Ethereal Monitoring Tool http://www.quepublishing.com/articles/article.aspx?p=22174

[9] R. Johnson & D. Wichern, “Applied Multivariate Statistical Analysis”, Prentice Hall International, Incorporated, ISBN-10: 0131877151, 607pp, 1988

[10] Multiple Non-linear Regression http://www.nlreg.com/?gclid=CKq367LH4boCFU5a3godRG0AxQ

[11] T. Kailath, A. H. Sayed, and B. Hassibi, Linear Estimation, Prentice Hall, NJ, ISBN 0-13-022464-2, 854pp, 2000.

[12] The MathWorks - MATLAB and Simulink for Technical Computing, http://www.mathworks.com/

Dr. Ioannis Barbounakis received an Electrical Engineering degree from the Aristotelian University of Thessaloniki, Greece (1991) and MSc and PhD degrees in Digital Communications from the Technical University of Crete, Greece and the University of Bradford, UK in 1993 and 2000 respectively. His dissertation thesis was in the area of Error Correction Techniques for the general class of Correlative Encoded Continuous Phase Modulation Signals. He has joined the Telecommunication Systems Institute of Crete (TSI) since its establishment in 1995. He is currently a Lecturer at the Department of Electronics, Technological Educational Institute of Crete. Dr. Barbounakis is also the director of the Computer & Information Technology Laboratory at the same Department. His research interests include GRID Middleware Technologies (Data Grids, Computing Grids), Grid Portal Toolkits and Frameworks, Efficiency and Thoughput of Data Networks, Microprocessor Applications in Telecommunications and Wireless Sensor Networks. He has been actively involved in various national and European collaborative research and development projects, with emphasis on Service Quality, Network Aspects & Grid Technologies. Dr. Barbounakis has published in refereed archival scientific journals, international conferences and workshops. He is co-author of the book chapter “Introduction to WLL: Digital Service Technologies” in “Wireless local Loops Theory and Applications” ISBN: 0471-49846-7, June 2001, Wiley & Sons, England and the book chapter "GRIDS in community settings", in “Virtual Community Practices and Social Interactive Media: Technology Lifecycle and Workflow Analysis”, ISBN: 978-1-60566-340-1, IGI Global Inc. He also serves as a reviewer for various archival journals.

Mr. Pantelis Askordalakis received his degree in Electronics from the Technological Educational Institute of Crete, Greece. He has been working on his Master degree, on Grid Computing. He has been actively participating in the establishment of the TECHGRID computing/networking facility under the Computer & Information Technology laboratory at Technological Educational Institute of Crete.

Dr. Michalis Zervakis holds a Ph.D degree from the University of Toronto, Department of Electrical Engineering, since 1990. He joined the Technical University of Crete on January 1995, where he has been serving as Professor at the department of

Electronic and Computer Engineering. He also joined the Telecommunication Systems Institute of Crete (TSI) since 2002. He served as Associate Editor in the "IEEE Transactions on Signal Processing" from 1994 to 1996. He was an assistant professor with the University of Minnesota-Duluth, USA, from September 1990 to December 1994. Prof. Zervakis is the director of the Digital Image and Signal Processing Laboratory (DISPLAY) at the Technical University of Crete. Under his direction, the lab is involved in research on modern aspects of signal processing, including estimation and constrained optimization, multi-channel and multi-band signal processing, wavelet analysis for data/ image processing and compression, neural networks and fuzzy logic with applications in biomedical data analysis, imaging systems and integrated automation systems. Developments also include DSP-based real-time implementation.

Dr. Demosthenes Akoumianakis is Professor at the Department of Informatics Engineering, Technological Education Institution of Crete (TEI-C), and the founder and Director of the interactive Software and Systems Engineering Laboratory (iSTLab, http://www.istl.teiher.gr/). He holds BA (Hons) in Computing in Business (1990) from the University of Huddersfield, UK, and MSc (1995) and PhD (1999) degrees in Human-Computer Interaction from the University of Kent at Canterbury, UK. Prof. Akoumianakis has been awarded the 'The High School Graduate's award' from the Municipality of Heraklion, Crete (1984) and the 1st IBM Prize for Final year Dissertation (The University of Huddersfield, 1990). Before joining TEI-C, he served as researcher at the Institute of Computer Science (FORTH), affiliated researcher at the University of Crete (ELKE) and systems analyst at the Co-operative Wholesale Society, Central Manchester Complex, UK. Prof. Akoumianakis is an active member of several research societies including ACM, Greek Special Interest Group on Computer-Human Interaction (SIGCHI) and serves as member of the International Editorial Board of IGI Global, member of the Executive Reviewers Board of the International Journal of Educational Technology & Society and member of the review committees of several international scientific journals and conferences.