Online State Estimation of Communication Networks

Atulya K. Nagar, Ghulam Abbas, and Hissam Tawfik

Intelligent and Distributed Systems Laboratory, Deanery of Business and Computer Sciences,
Liverpool Hope University, Liverpool, L16 9JD. UK.

Summary
This paper introduces a novel approach to congestion avoidance in TCP/IP networks. The new technique is based on State Estimation which addresses some general shortcomings of the current Active Queue Management schemes such as RED. State Estimation of dynamic systems is an important requirement for secure and economical process operations. It is an intrinsic element of many network management systems including Power, and Water Distribution Networks, where its implementation not only facilitates real-time network monitoring, fault detection, and process optimization, but it also enables an advanced control with improved system security. This paper presents some potential issues in TCP/IP networks where State Estimation can be applied to achieve better performance in congestion control. The results presented in this paper prove that there exist many uncertainties, anomalies, bad data, and measurement noises, in a real physical network, which if not remedied, can affect the efficiency of the queue management to a greater extent. The State Estimation scheme, proposed in this paper, is capable of filtering out the noise and hence provides an optimal control and security. This is validated by comparing the results to the simulation results of one such RED algorithm.

Key words:
Active Queue Management; filtering; State Estimation; TCP/IP networks, Kalman Filters, Gaussian noise.

1. Introduction

Congestion typically refers to a situation when a TCP service either fails to fulfil a request to transfer a bulk of data, or it ends up with extensive service delays. Furthermore, data packets may also be lost in an attempt to complete the request. If the congestions are not dealt with appropriately, the packet loss rate becomes high enough, giving rise to retransmissions of lost packets and consequently cause further service delays. The Transmission control protocol (TCP) has been designed exclusively to offer a reliable service in terms of data delivery. Early implementations of TCP led to, what was known as “congestion-collapse”, in which a network failed to respond altogether. This situation was soon overcome by many reliable TCP implementations [15, 16]. However, the rapid increase in users around the globe, with a consequent increase in data requirements, has offered many threats to this reliability. Many approaches have been developed over the last few decades of TCP history and the modelling and simulation of congestion avoidance has been improved considerably. These approaches either include scheduling mechanisms to manage the network resources, or provide techniques to avoid congestions by implementing flow control and provide policies for the queue management. These include Random Early Detection gateways (RED) [12] and further improvement to RED [10, 11, 17, 18, 28]. There are packet level models [1], fluid based models [13, 20], and hybrid models [3]. But most of these techniques do not focus on modelling the uncertainties or noises that may be present in the physical system. These limitations are studied and remedied in this paper, by applying State Estimation mainly due to the following reasons. At first, no mathematical model is perfect and therefore may not capture all behavioural aspects of the actual physical state of the system. Numerous effects of the underlying system are deliberately left un-modelled, while the assumptions of the modelled effects are not correct under all circumstances. As such, there may be many uncertainties present in any mathematical model. Moreover, the underlying systems are driven not only by the control inputs, but are often driven by disturbances or noises which cannot be modelled deterministically. This can certainly affect the outputs which do not reflect the exact system state and cause further uncertainties when used as a feedback control, as will be shown in examples. Furthermore, many measurement devices on the network may also be noise corrupted, and as such the received measurements do not provide perfect and complete data about the system.

Two implementations of State Estimation have been studied during this research, which have provided the motivation for applying a similar concept to TCP congestion control. State Estimation was applied to The Power Distribution Networks in late 1960’s. Before that time the Power management systems had been facing many threats to their system security and control. The implementation of State Estimation provided a solution to these problems and today State Estimation is an essential element of a Power Management System where it is used to "fine-tune" system state variables by minimizing the sum of squares of the differences between the estimated and measured power system data e.g. current, voltage, resistance etc. The measured data may be affected by errors e.g. due to meter inaccuracies, and it is impossible to use meters to measure every power system state. The State Estimator estimates all the pertinent state variables by using measured data, based on the user's models and provides real time measurements for online monitoring [9, 25, 31]. A similar concept was implemented in the Water Distribution Networks, where a limited number of measurements, e.g. flow and pressures are measured directly from the system. These measurements are then used by water system State Estimators together with the knowledge of network topology to provide real-time measurements of the complete system state. Different filtering techniques are used to develop State Estimators which are also used to reduce the measurement noise present in the system, detect errors and control the online system monitoring [21, 22, 23, 24].

The State Estimation technique proposed in this paper uses the Kalman filtering approach to try to address the general problems of RED models. We have used the discrete time model proposed by [13], as a case study in this paper to compare the results from our State Estimator to the Simulation results of this model.

2. TCP Traffic Simulator

A simulator is designed and implemented in MATLAB using the congestion avoidance model presented in [13]. A brief description of the process operation is presented here before proceeding towards our ultimate goal of State Estimation. The basic mechanism to control the flow of data, used by TCP, is called *congestion window* $\boldsymbol{W}$, which is the TCP state variable that limits the amount of data a sender can transmit at a given instance [27]. The TCP state variable that is used to control the length of queue at a router's buffer is called queue length $\boldsymbol{q}$ [4]. The exponentially weighted average queue length $\boldsymbol{X}$ is a state variable that limits the flow control in an RED [13]. The time elapsed between the departure of a packet from the sender and the return of the acknowledgment to the sender is called Round Trip time $\boldsymbol{T}$ [13]. During congestion avoidance (additive increase) the congestion window $\boldsymbol{W}$ of each sender grows linearly causing a linear increase in the queue size. This growth continues and the system operates under congestion avoidance phase as long as $\boldsymbol{X}$ remains below $q_{\min}$. Once $\boldsymbol{X}$ exceeds $q_{\min}$, a drop is assumed to have occurred, after which the systems is transitioned to the delayed drop-notification phase. This phase is governed by a count down timer T_k which expires in exactly one round trip time m_k . During this phase the system obeys the same set of equations, it obeyed in the previous phase to update each state variable. This phase ends when the timer T_k , after which the congestion window $\boldsymbol{W}$ is cut in half and a new timer T_k , is used which is initialised to the value of the current round trip time. The system transitions to the third phase called recovery. During this phase the sender doest not transmit any packets, the congestion window $\boldsymbol{W}$ is kept fixed and the queue is emptied at each time step. This phase lasts for .5 m_k time steps after which the system transitions to the last phase which also lasts 0.5 m_k time steps. The sender is now allowed to transmit packets but the congestion window $\boldsymbol{W}$ is still kept fixed resulting in no change in queue length. The model returns to the congestion avoidance phase once this timer (0.5 m_k) expires. The designed simulator assumes the network configuration shown in figure 1 and uses the parameters of table 1. Figure 2 and 3 presents the simulation results for two and three senders respectively in order to elaborate the changes that occur in the TCP state variables W, q and X during the congestion avoidance process.

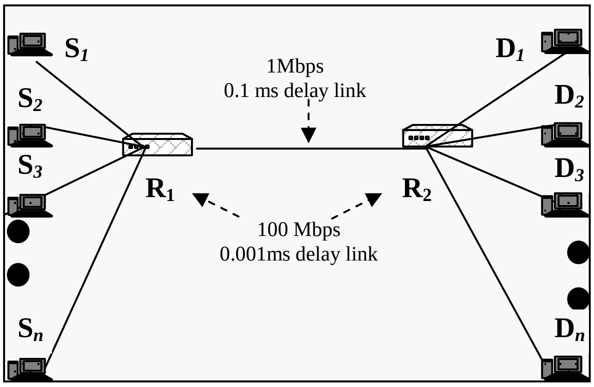

Figure 1 Dumbbell Topology

Table 1 Network parameters

Variable	**Description**	**Value**
q_{min}	RED parameter	150
q_{max}	RED parameter	300
T_p	Propagation delay	0.1s
B	Bandwidth of bottleneck link	1Mbps
p_{max}	RED parameter	0.1
Weight	RED parameter	0.001

3. TCP Traffic State-Estimation Problem

The general TCP traffic flow State Estimation problem can be posed in a similar way as formulated in the Power and Water systems. The concept is similar to the State Estimation of the two networks (Power & Water), however, the physical construct is different in the case of TCP and therefore a different approach is required. To have a more concrete description of the problem, consider the following. Let x_k be a given signal at time step k and E be the noise. Considering that only the sum of signal and the noise can be observed, it can be generally represented as,

$$\boldsymbol{Z} = HX + \mathrm{E} \tag{1}$$

Where, $\boldsymbol{Z}$ is the measurement vector which is updated at each scan. X is the State vector, H is as Identity matrix ($m \times n, m \geq n$) relates state to measurement Z, and E is the Vector of measurement errors.

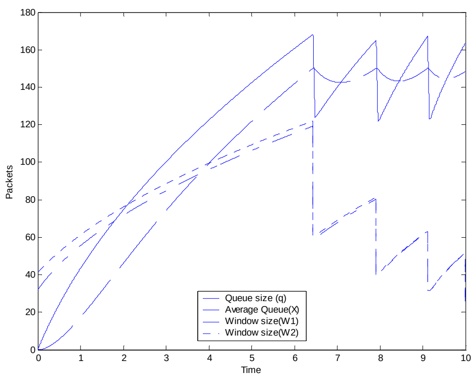

Figure 2 Two senders

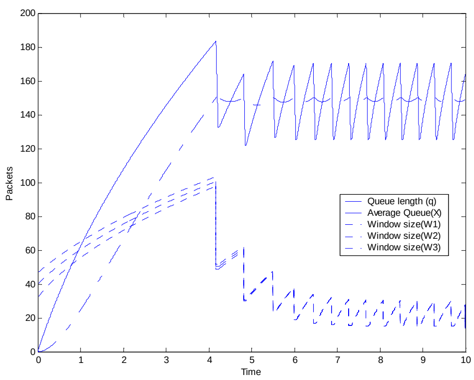

Figure 3 Three senders

The error E can arise due to a number of situations, e.g., inaccuracy of network model, measurement noise and inaccuracy of RED. The mathematical model may serve its purpose well in most cases but the assumptions of the mathematical model are not correct in all circumstances, for example, presence of a very large number of network nodes and packet-flows may affect the calculations up to a fractional level which can accumulate into a high level of measurement uncertainty when the results are used as feedback control. Moreover, an RED that operates on a router at some congested link and uses the exponentially-weighted-average-queue-length to predict packet losses and impose flow control may use wrong parameters (weights) and can consequently lead to uncertainties.

The aim of State Estimation is to separate the signal of interest from the measurement uncertainties and filter out the noise. A Kalman filter State Estimator is used here for this purpose. The state variable whose measurements are the elements of vector Z can be the congestion windows $\boldsymbol{W}$ of each sender in the underlying congested connection, queue length $\boldsymbol{q}$ of the underlying router and the average queue length $\boldsymbol{X}$.

4. The TCP Traffic State-Estimator

We first derive expressions for the time and measurement updates of the Kalman filter [29], in order to devise a State Estimation algorithm. The state vectors to be estimated are the congestion window $\boldsymbol{W}$**,** the queue length $\boldsymbol{q}$ and the average queue length $\boldsymbol{X}$. The general stochastic differential equation that governs the discrete time controlled Kalman process [19, 29] is given as

$$x_k = Ax_{k-1} + Bu_k + y_k \tag{2}$$

The process model we used [13] consists of four phases where the congestion window state variable $\boldsymbol{W}$ in each phase is governed by a separate differential equation. In order to derive an equivalent expression for the above equation (2), let C_k denotes the relative equation of the process, so it can be represented as,

$$C_k = \begin{cases} W_k + 1/W_k & \textit{Congestion Avoidanve} \\ W_k + 1/W_k & \textit{Delayed Notification} \\ W_k & \textit{Recovery (Not Sending)} \\ W_k & \textit{Recovery (Sending)} \end{cases}$$

Also, there is no control input in the system while controlling the congestion window $\boldsymbol{W}$, therefore, $u = 0$ .Equation (2) can be written to represent $\boldsymbol{W}$ as,

$$\boldsymbol{W}_k = C_{k-1} + y_k \tag{3}$$

The measurement equation (1) can be written for $\boldsymbol{W}$ as,

$$\boldsymbol{Z}_k = H\boldsymbol{W}_k + v_k \tag{4}$$

The noisy measurements are of the system directly (H as identity [19]), therefore,

$$\boldsymbol{Z}_k = \boldsymbol{W}_k + v_k \quad (5)$$

The process noise y_k and the measurement noise v_k, are assumed to be white noises, independent of each other, and have a Normal Distribution.

$p(v) \sim N(0, R)$

$p(y) \sim N(0, Q)$

The above expression can now be used to derive the time and measurement update equations, for the Kalman Filter

Time Update	**Measurement Update**
$\bar{\boldsymbol{W}}_k = \boldsymbol{W}_{k-1}$	$K_k = \bar{P}_k(\bar{P}_k + R)^{-1}$
$\hat{P}_k = P_{k-1} + Q$	$\hat{\boldsymbol{W}}_k = \bar{\boldsymbol{W}}_k + K_k(\boldsymbol{Z}_k - \bar{\boldsymbol{W}}_k)$
	$P_k = (1 - K_k)\bar{P}_k$

The process equation for the exponentially average queue-length $\boldsymbol{X}$ remains the same for all four phases of the congestion avoidance process in RED, therefore its implementation in the State Estimator is straightforward, and the same algorithm can be used for $\boldsymbol{X}$ with appropriate H .

The process for the queue length $\boldsymbol{q}$ can be represented as:

$$C_k = \begin{cases} q_k + 1/W_k & \text{Congestion Avoidanve} \\ q_k + 1/W_k & \text{Delayed Notification} \\ q_k - 1 & \text{Recovery (Not Sending)} \\ q_k & \text{Recovery (Sending)} \end{cases}$$

The process equation (2) in this case becomes

$$\boldsymbol{q}_k = C_{k-1} + y_k$$

The measurement equation (1) for the queue length $\boldsymbol{q}$ can be written as:

$$\boldsymbol{Z}_k = \boldsymbol{q}_k + v_k \quad (6)$$

The time and measurement update equation in case of $\boldsymbol{q}$ can now be written as,

Time Update	**Measurement Update**
$\bar{\boldsymbol{q}}_k = \boldsymbol{q}_{k-1}$	$K_k = \bar{P}_k(\bar{P}_k + R)^{-1}$
$\hat{P}_k = P_{k-1} + Q$	$\hat{\boldsymbol{q}}_k = \bar{\boldsymbol{q}}_k + K_k(\boldsymbol{Z}_k - \bar{\boldsymbol{q}}_k)$
	$P_k = (1 - K_k)\bar{P}_k$

5. Results

The derived Kalman algorithm for the queue length $\boldsymbol{q}$**,** average queue length $\boldsymbol{X}$ and congestion window $\boldsymbol{W}$ is implemented in MATLAB. The following section presents comparison of the results from the Simulator [13] and the Estimator using the dumbbell topology [3], and the configurations of the congested connection listed in table 1.

5.1 Estimating Congestion Window

According to the law of flow conservation [3] the flow into a congested link depends on the number of packets being injected by a sender into a link and as such, the accuracy of congestion window size is of significant importance. The congestion avoidance model [13] used in this paper, increments congestion window by ***1/W*** after the receipt of each acknowledgment. While this could work well for a small number of senders, it can lead to uncertainties in the presence of a large number of senders simultaneous transmitting through a link. As the acknowledgement is modelled to arrive in one round-trip time ($T_p + q/B$), which depends on the queue length $\boldsymbol{q}$ (queuing delay, $\boldsymbol{q/B}$), the estimation assumes the round trip time to be corrupted by a small fraction (0.0005s). This fractional change is certain to occur when there is some background traffic present i.e. the congested router is also serving some other flows arriving from other nodes (note that the model assumed no background traffic). This fractional inaccuracy in the round-trip time can accumulate into large inaccuracy and consequently, the congestion window measurement becomes noisy after a few round trip times. For example, assuming 10 senders simultaneously transmitting through a queue and then calculating the round-trip time, the noise robustness becomes 0.0067s. This could have a dramatic affect on the process operations as will be illustrated in the following sections. Running the estimator by using the configurations listed in table 1 for the dumbbell topology (figure 1), produces the result shown in figures 4 through 8, where the noisy measurement are the measurement from the Simulator [13]. The noise robustness increases at the start of the congestion soon after the first packet drop. This is because the sender waits for a relatively longer round trip time during the recovery phase (note the horizontal increase of the noisy measurements). Noise robustness also increases with the more increased number of senders. Figure 4 compares the result of simulator [13] and estimator for four senders (Plotted only one sender's window for clarity).

The difference between the noisy and estimated measurements can be better observed by eliminating the initial additive-increase phase and then re-plotting the result for the last two seconds of each run. These are represented by the relative figure-number and an extension A in the following section.

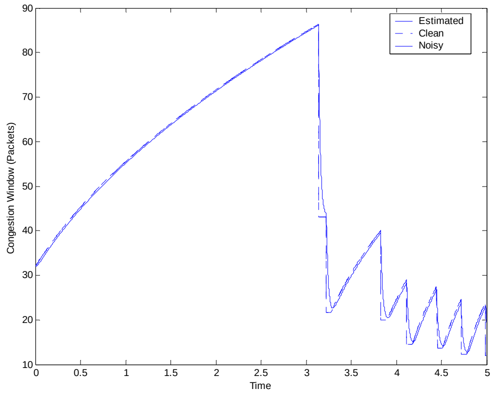

Figure 4 Performance comparisons for four senders

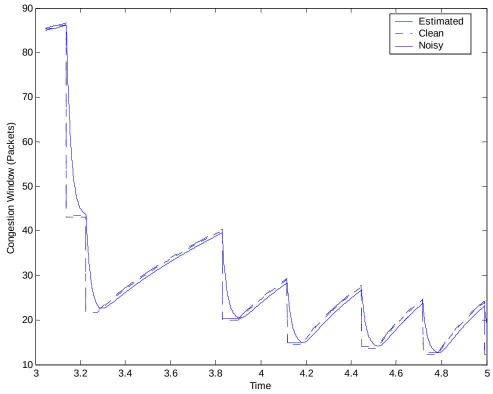

Figure 4A Recompression

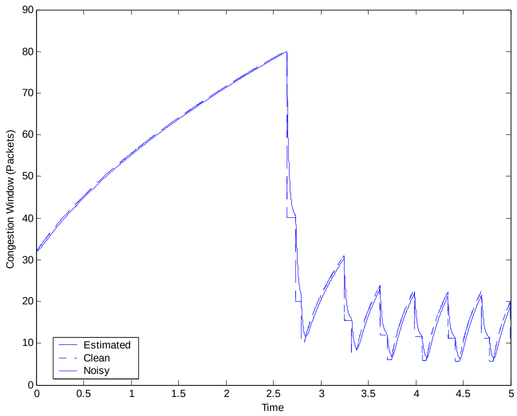

Figure 5 Performance comparisons for five senders

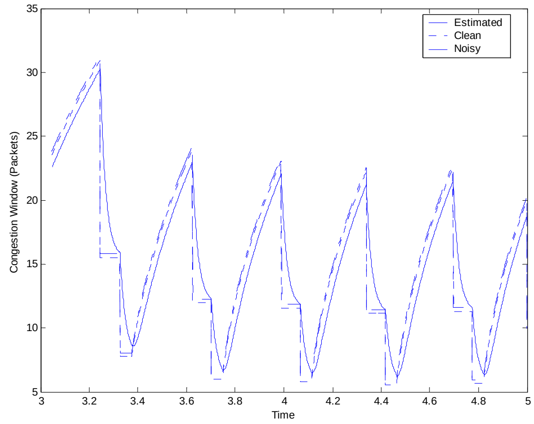

Figure 5A Recompression

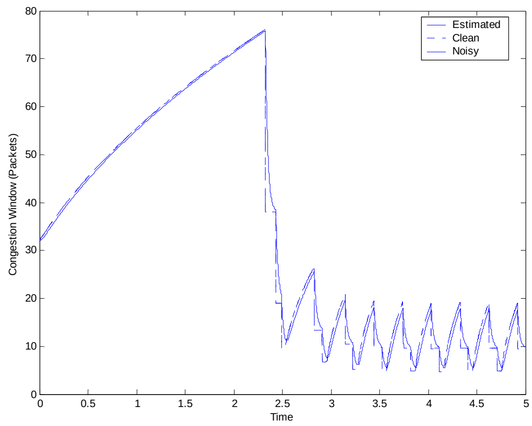

Figure 6 Performance comparisons for six senders

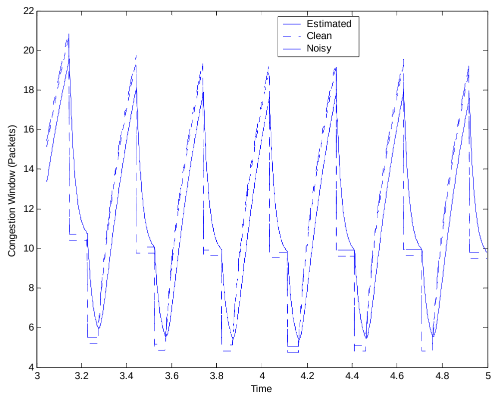

Figure 6A Recompression

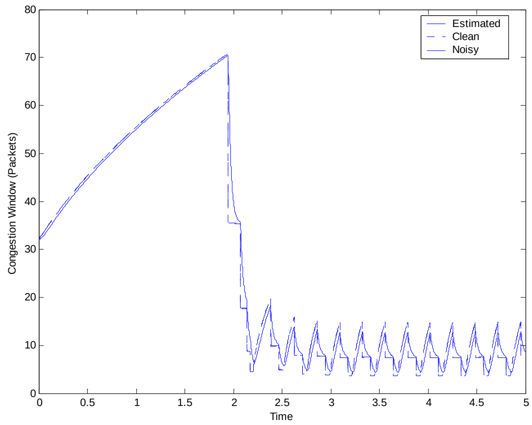

Figure 7 Performance comparisons for eight senders

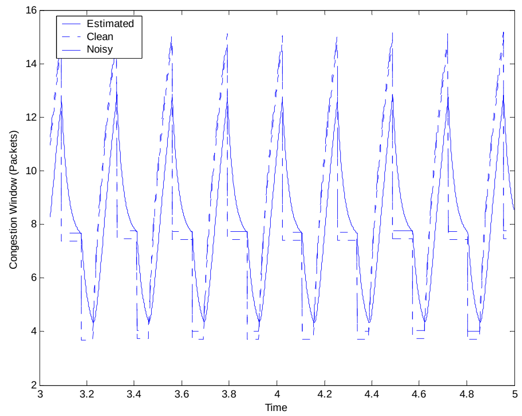

Figure 7A Recompression

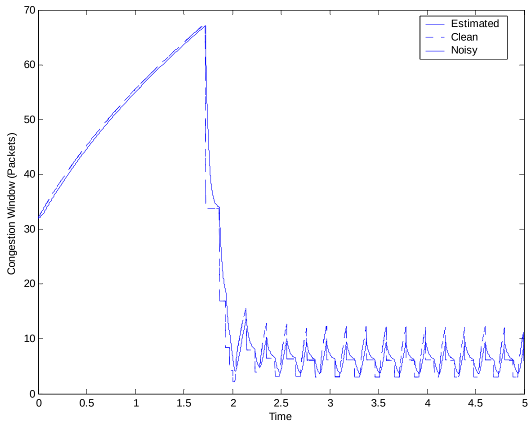

Figure 8 Performance comparisons for ten senders

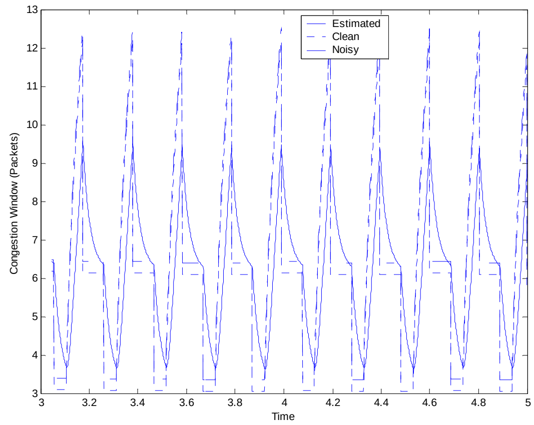

Figure 8A Recompression

Table 2 Noise Robustness

Noise Robustness			
No. of Senders	Simulated Mean Window size (without noise)	Simulated Mean Window size (with noise)	Estimated Mean Window size (with noise)
4	48.8446	50.6776	48.9165
5	39.2134	41.9166	39.3013
6	33.7183	36.0158	33.8547
8	26.9072	30.1833	26.9175
10	23.0093	27.9180	23.1886

5.2 Estimating Queue Length

This section presents the results of estimated queue length which are compared to the results of simulated queue length [13] in the presence of certain noise. The queue length is observed for a number of senders, simultaneously transmitting through this queue. After adding the noise to the signal, each observation was made for ten seconds as shown in the following figures (9 through 12). As the measurement of the queue length depends on the congestion window size of each sender transmitting through this queue $q = 1/W$, the estimator assumes five senders transmitting simultaneously, and the corrupted value of each $\boldsymbol{W}$ shown in table 2, the flow

$q = \sum 1/W$ can produce the following effects on the queue size.

In the following figures each observation is made for 10 seconds for various numbers of senders. The results are redrawn in MATLAB were necessary for the last two seconds of each run in order to improve the visibility of the difference between the noisy and estimated measurements.

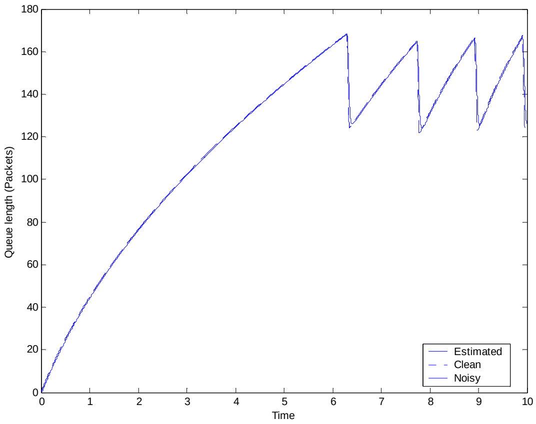

Figure 9 Performance comparisons for two senders.

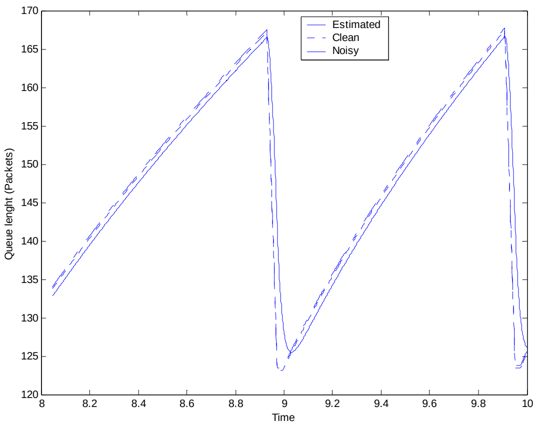

Figure 9A Recompression

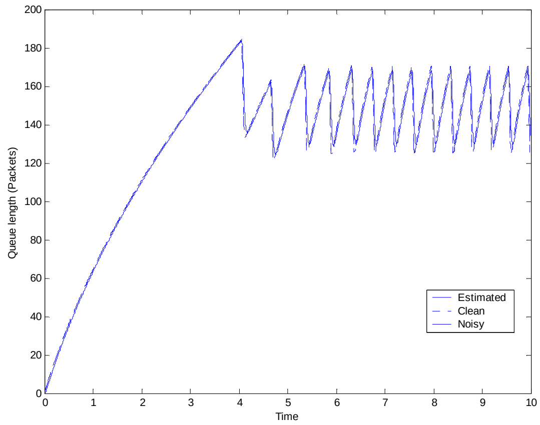

Figure 10 Performance comparisons for three senders

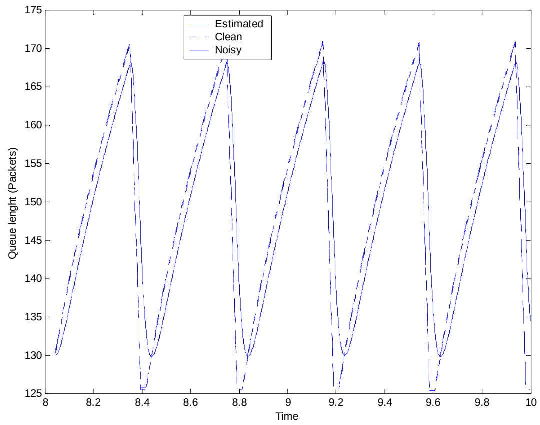

Figure 10A Recompression

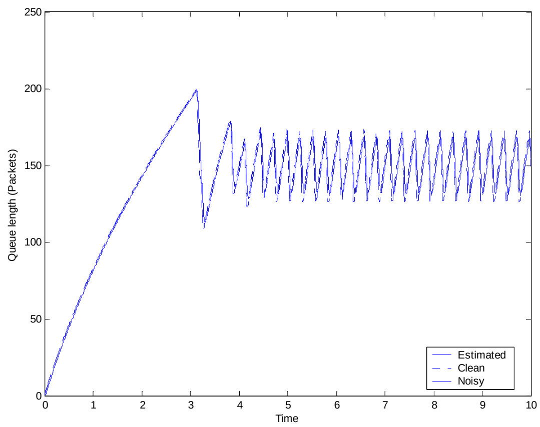

Figure 11 Performance comparisons for four Senders

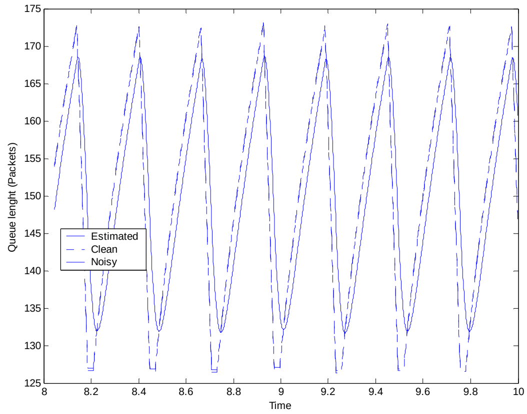

Figure 11A Recompression

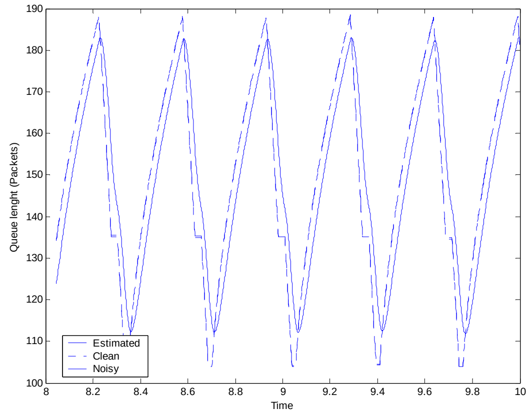

Figure 12A Recompression

Table 3 Noise Robustness

Noise Robustness			
No. of Senders	Simulated Mean Queue-Length (without noise)	Simulated Mean Queue-Length (with noise)	Estimated Mean Queue-Length (with noise)
2	116.6457	119.3910	116.6333
3	131.3254	137.9962	131.3130
4	137.8840	150.1776	137.8696
5	141.0444	160.0377	141.0266

Table 3 lists mean queue-length of each sender observed for 10 seconds. In the presence of noise, the measurement of the simulated queue length reflects wrong values. It means that, some of the buffer space at the router's queue remains unutilized due to the noisy measurements, and the packets are dropped by the router (when queue reaches its capacity) whereas, in reality, there still remain some unoccupied space. The results from the estimator match closely with the results of simulator which assumes no noise, i.e. the estimation is capable of removing the measurement noise and reflecting correct mean queue-length. Moreover, it is also clear from the results that the measurement inaccuracy increases with the increased number of senders, consequently more packet drops will occur which in turn will lead to increased level of congestion. Figure 13 shows the variations in queue length and the comparison of estimated and noisy measurements for twenty senders.

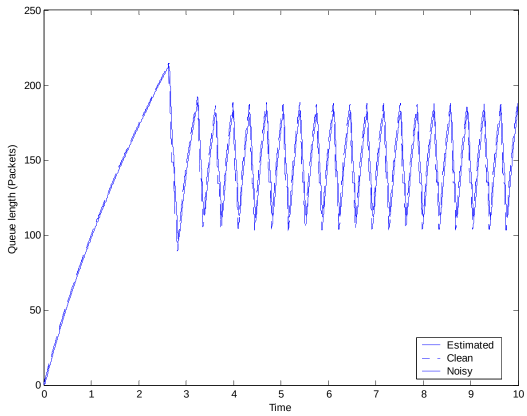

Figure 12 Performance comparisons for five senders

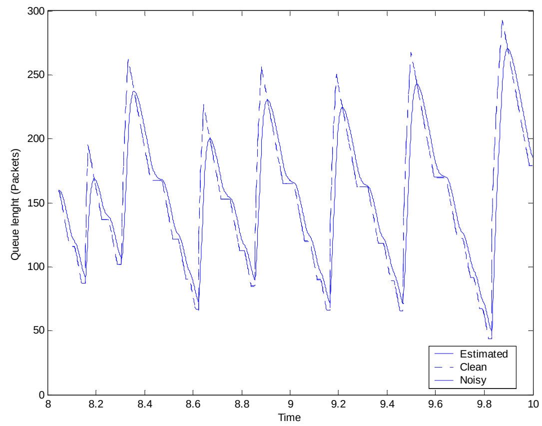

Figure 13 Performance comparisons for twenty senders

5.3 Estimating Average Queue

A router using RED will drop packets as soon as $\boldsymbol{X}$ reaches $\boldsymbol{q}_{min}$. As the measurement of the average queue depends on the actual queue length $\boldsymbol{q}$, the noisy measurements of $\boldsymbol{q}$ can in turn affect the measurement of $\boldsymbol{X}$ resulting in early packet-drops prior to buffer filling. A State Estimation coupled with RED algorithm can thus provide a better control of the system. Figure 14 shows a comparison of the noisy and estimated measurements of $\boldsymbol{X}$ in a three sender scenario. The result shown in Figure 14 can be better analyzed when viewed over a short time scale, as shown in Figure 14A**.** It is clear from the figure 14A that the noisy measurements reach $\boldsymbol{q}_{min}$ relatively early and causes an early packet drop, which can affect the system control when large numbers of connections are present. Figures 15 through 19 show re compressions over short time scales for increased number of senders.

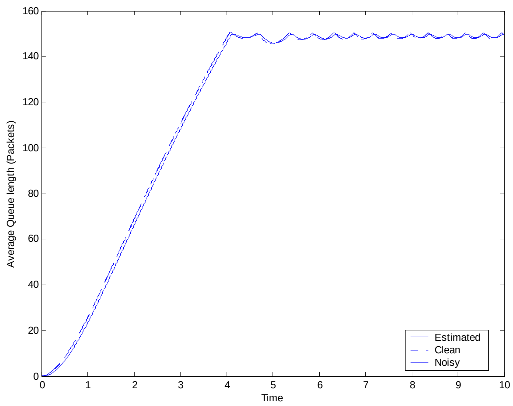

Figure 14 Performance comparisons for three senders

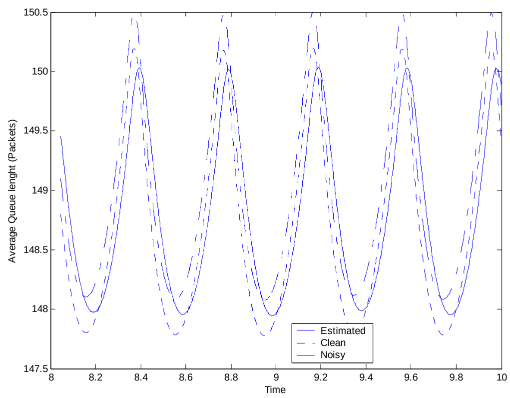

Figure 14A Re-comparison using smaller timescale (three senders)

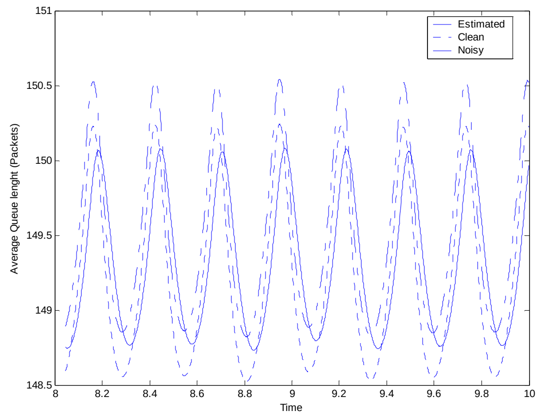

Figure 15 Re-comparison using smaller time scale (four senders)

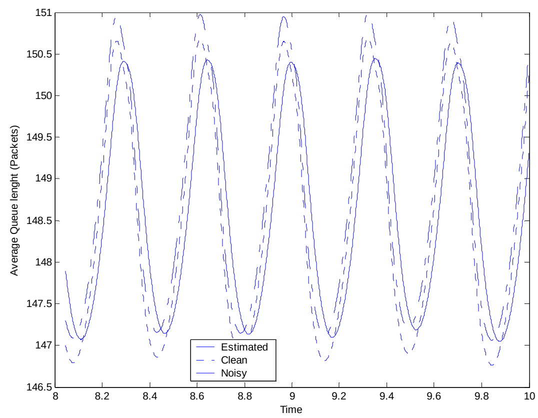

Figure 16 Re-comparison using smaller time scale (five senders)

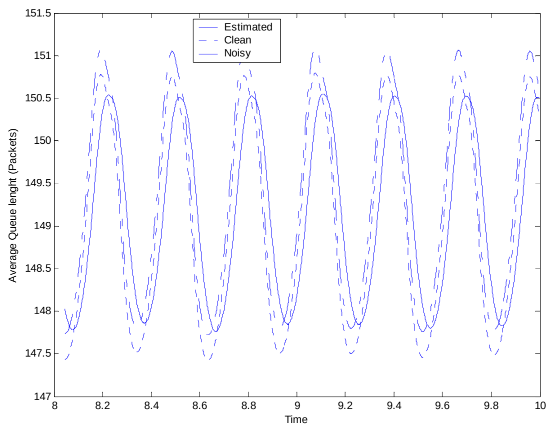

Figure 17 Re-comparison using smaller time scale (six senders)

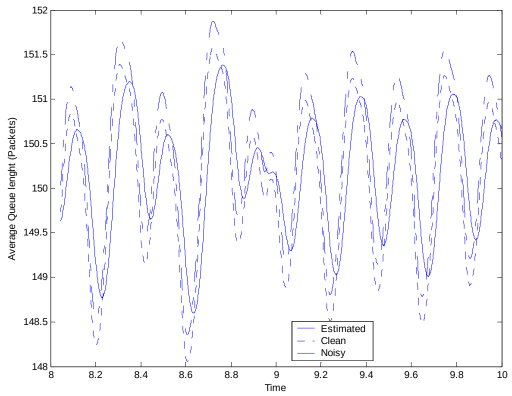

Figure 18 Re-comparison using smaller time scale (ten senders)

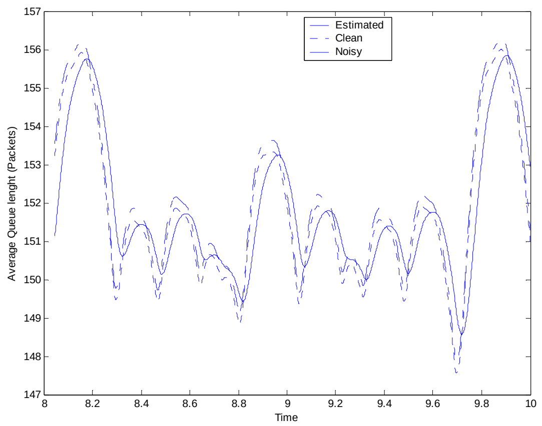

Figure 19 Re-comparison using smaller time scale (twenty senders)

6. Discussions

A very accurate modelling technique can help avoiding congestion to a greater extent. Congestions can even be eliminated completely when the modelled network does not have any uncertainties, anomalies, bad data, and measurement noises, in its physical operations. As this is not possible in reality, even the most accurate and fastest modelling and simulation technique can not guarantee a complete elimination of congestions. While these anomalies can be negligible for a small network as the created congestion may not be even observed by the users, examples have shown that a small fraction of noise can have a dramatic affect over the packet delivery in the networks serving heavy loads of users. A congestion avoidance modelling technique coupled with State Estimation can therefore be worthwhile as it is capable of eliminating the anomalies and consequently the congestions.

The congestion window W is the amount of data (packets) a sender can transmit at a given instance of time. The congestion avoidance model [13] keeps incrementing W by $1/W$ each time an acknowledgment is received, until the occurrence of congestion (first packet drop), The acknowledgment is modelled to arrive in one round trip time ($T_p + q/B$). Also, during all other congestion control phases of the model (delayed drop notification, recovery [not sending] and recovery [sending]), the size of W and the amount of packets being sent depend on the arrival of acknowledgments (each acknowledgment in a round trip time, $T_p + q/B$). This will only return the exact value of round trip time when there is no background traffic (as the model ignored the background traffic), whereas in reality, this rarely happens as the network may be serving other traffic flows and, as such, we assume the calculated round trip time value to be corrupted by a small fraction. In reality this fractional error may cause fractional delays in data delivery which may be negligible, but for a large number of flows present in a network, this may intensify the congestion. The Estimator on the other hand is able to keep track of the round trip time values and eliminate any fractional changes that may arise during the measurements. A comparison of the performance of both the simulator and the estimator is presented in figures 4 through 8, where the noisy measurements of W cause relatively large amount of packets to be injected in the network.

The queue at a router forms the basis of the entire congestion avoidance process of a modelled network. The model keeps track of the queue length q and when the queue reaches it capacity, congestions occur. As such, the model must keep track of the queue length most accurately, in order to avoid congestion. The measurement of q depends on the amount of packets being injected in a router's queue which in turn depends on the congestion window W of each sender transmitting through this router $(q = q+1/W)$. The corrupted values of W (table 2) will cause more packets to be injected in the queue causing further congestions. Also, the measurement noise in q will reflect wrong values of the queue length. The estimator is capable of reducing the noise and hence maintains the flow control (Table 3).

The accuracy of the measurements of average queue length X are of significant importance in terms of systems control, as such the average queue length X plays a vital role in any RED algorithm as it is used to keep track of the actual queue length, and depending on its value the packets are dropped (e.g.

when $\boldsymbol{X}$ reaches q_{min} =150 packets, a drop occurs [13]. The presence of noise in $\boldsymbol{W}$ and $\boldsymbol{q}$ eventually will cause wrong calculation of $\boldsymbol{X}$, resulting in early packet-drops (Figure 14 through 19). The performance of a RED algorithm for a large number of flows can suffer in the presence of a large number of flows. The superior performance of the estimator can be seen in figures where the value of $\boldsymbol{X}$ remains unaffected even in the presence of noisy conditions

7. Conclusions

The results have shown that the application of State Estimation in RED can serve many control and security benefits. However, the State Estimation is expected to covers many more areas as it does in Power and Water Distribution Systems. There is a need to construct a complete network management system that will help to decompose the network into smaller *observable-islands,* in order to further improve the observability and controllability of the networks like WAN. This will also help in ensuring the feasibility of the system (to answer the questions like: if observable, how observable is the system?), before a state estimator is run (physical implementation). Current State Estimator only assumes Gaussian noise in the measurement data, however, in physical systems; noise distribution is not always Gaussian. Also, there might be uncertainties and bad data associated with the actual network model (e.g. missing nodes, broken links). Uncertainty modelling and its analyses is therefore another proposal for further research.

References

[1] Altman, E., Avrachenkov, K. and Barakat, C. (2005) "*A Stochastic Model of TCP/IP With Stationary Random Losses*", IEEE/ACM Transactions on Networking(TON), vol. 13, no. 2, p.356-569.

[2] Bogoyavlenskaya, O. Y. (2004) "*Teaching Networking Congestion Control*", in proceedings of 9th Annual Joint Conference on Innovation and technology in Computer Science Education, Leeds, UK, 28-30 June, v. 36, no. 4, p. 35-41.

[3] Bohacek, S., Hespanha, J., Lee, J., and Obraczka, K. (2003) "*A Hybrid System Modeling Framework for Fast and Accurate Simulation of Data Communication Networks*", in proceedings of the International conference on Measurement and modelling of computer systems, San Diego, California, USA, 10-14 June, p.58-69.

[4] Braden, B., Clark, D., Crowcroft, J, Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Partridge, C., Minshall, G., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., Zhang, L. (1998) "*Recommendation on Queue Management and Congestion Avoidance in the Internet*", RFC 2309, April 98.

[5] Brakmo, L. S., O'Malley, S. W. and Peterson, L. L. (1994) "*TCP Vegas: New Techniques for Congestion Detection and Avoidance*", in proceedings of the Annual Conference on Communications Architectures, Protocols and Applications**,** London, UK, 31 August – 02 September, vol. 24, no. 4, p.24-35

[6] Comer, D. E. and Stevens, D. L. (1999) "*Internetworking with TCP/IP, design, implementation and internals*", Third Edition, Volume II, Prentice Hall, New Jersey

[7] Comer, D. E. (2000) "*Internetworking with TCP/IP, principles protocols and architectures*", Fourth Edition, Volume 1, Prentice Hall, New Jersey.

[8] Curran, K., Woods, D., McDermot, N., and Bradley, C. (2003) "*The effects of badly behaved routers on the internet congestion*", International Journal of Network Management, January 2003, vol. 13, no.1, p. 83-94.

[9] Dy Liacco Corporation, (1990) "*The role and implementation of state estimation in an energy management system*", A special Report of EPES, Electrical Power& Energy Systems, Cleveland, 2 April, vol.12, no. 2, p.74-79.

[10] Feng, W., Kandlur, D., Saha, D. and Shin, K. G. (2001) "*BLUE: An Alternative Approach to Active Queue Management*", in proceedings of the 11th International workshop on Network and Operations Systems, 25-26 June, Port Jefferson, New York, USA, p.41-50

[11] Feng, W., Kandlur, D., Saha, D. and Shin, K. G. (1997) "*Techniques for Eliminating Packet Loss in Congested TCP/IP Networks*", Technical Report, IBM Watson Research Centre, November 1997, vol. 1, p.1-21.

[12] Floyd, S. and Jacobson, V. (1993) "*Random Early Detection Gateways for Congestion Avoidance*", IEEE Transactions on Networking, New York, United States, August 1993, Vol. 1, no. 4, p. 397-413

[13] Frommer, I., Harder, E., Hunt, B., Lance, R., Ott, E. and Yorke, J. (2004) "*Two Models for the Study of Congested Internet Connections*", international Conference on Communications and Computer Networks, (CCN 04), Proceedings of the IASTED, 13 September, Cambridge, MA

[14] Hart, C. W. (1998) "*On-Line first principle models – State estimation*", in proceedings of IChemE Fifth Conference on Advances in Process Control, 2-3 September, Swansea.

[15] Jacobson, V. (1988) "*Congestion avoidance and control*", in symposium proceedings on Applications, Architecture and Protocols for Computer Communications, Stanford, California, US., 16-18 August, vol. 18, no. 4, p.314-329

[16] Jacobson, V., (1990) "*Modified TCP Congestion Avoidance Algorithm*", Technical Report, Network Research Group LBL.

[17] Mathis, M. and Mahdavi, J. (1996) "*Forward Acknowledgements: Refining TCP Congestion Control*", in proceedings of the International Conference on Applications, architectures, and protocols for computer communications, California, United States, 28-30 August, Vol. 26, no. 4, p. 281-291

[18] Mathis, M. Mahdavi, J., Floyd. S., and Romanow. A. (1996) "*TCP Selective Acknowledgment Options*", RFC 2018, October 96.

[19] Maybeck, P.S. (1979) "*Stochastic models, estimation, and control*", Volume 1, Academic Press, London

[20] Misra, V., Gong, W., and Towsley, D. (2000) "*Fluid based Analysis of a Network of AQM Routers Supporting TCP Flows*", with an Application to RED, Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, proceeding of Sigcomm2000, 28 Aug– 01 Sep, Stockholm, Sweden.

[21] Nagar, A. K. and Powell, R. S. (2001) "*LFT/SDP approach to the uncertainty analysis for state estimation of water distribution systems*", IEE Journal of Control Theory and Applications, March 2002, vol.149, issue. 2, p.137-141.

[22] Nagar, A. K., and Powell, R. S. (2000) "*Observability Analysis of Water Distribution Systems under Parametric and Measurement Uncertainty*", Technical Report, in proceedings of ASCE Joint Conference on Water Resources Engineering and Water Resources Planning and Management, 12 July, Minneapolis, US

[23] Powell, R. S., Andersen, J. H., Nagar, A. K. and Hindi, K. S. (2000) "*State Estimation and Uncertainty Analysis in Water Networks*", Technical Report, in Proceedings of Workshop 2, EPSRC, UKWIR Project on Monitoring, Modelling & Leakage Management in Water distribution Networks, December 2000, London, UK

[24] Schrader, B. P. and Moore, S. F. (1977) "*Kalman Filtering in Water Quality Modeling*", Theory Vs Practice, in proceedings of the 9th Conference on Winter Simulation, 5-7 December, Gaithersburg, Maryland, United States [available at – portal.acm.org, April 2005]

[25] Slutsker, I. W. (1988) "*Bad Data identification in Power System State Estimation based on measurement Compensation and Linear Residual Calculation*", IEEE Transactions on Power Systems, IEEE Power System Engineering Committee, New York, United States, 31 January -05 February, vol.4, no. 1, p. 53-59.

[26] Stevens, W. R. (1996) "*TCP/IP Illustrated*", The Protocols, Vol.1, Addison-Wisley, USA.

[27] Stevens, W. (1997) " *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*", RFC 2001, January 1997.

[28] Stewart, R. R. and Xie, Q. (2001) "*Stream Control Transmission Protocol*", STCP, A Reference Guide, Addison-Wesley.

[29] Welch, G. and Bishiop, G. (2001) ""*An Introduction to the Kalman Filter*", Technichal Report, SIGGRAPH 2001, Los Angeles Conventional Contre, 12-21 August.

[30] Woods, R, L. and Lawrence, K, L. (1997) "*Modeling and Simulation of Dynamic Systems*", Prentice Hall, Inc. New Jersey.

[31] Wu, F. F. (1990) "*Power System State Estimation: A Survey*", International Journal of EPES, Electric Power & Energy Systems, 2nd April, p. 80-87.

ATULYA K. NAGAR is an Associate Professor in Computing Science and leads the Intelligent and Distributed Systems (IDS) Research Group at Liverpool Hope University, UK. He earned his PhD in Applied Mathematics from the University of York in 1996 and was a recipient of the prestigious Commonwealth Fellowship during 1993–1997. He holds a BSc (Hons.), MSc, and MPhil (with distinction) from the MDS University of Ajmer, India. He has worked for several years at Brunel University, UK. He has multidisciplinary research expertise in: systems engineering, computational biology, applied informatics, and applied nonlinear analysis.

GHULAM ABBAS received his MSc in Distributed Systems (2005) from Liverpool Hope University, UK and holds a BCS in Computer Science from Edwards College Pakistan (2003). Ghulam is currently embarking on his PhD and his research interests are in modelling, simulation, and distributed systems and networking technologies.

HISSAM TAWFIK is a Senior Lecturer in Computing Science at Liverpool Hope University. He holds an MSc and PhD from UMIST. His research areas are in Artificial Intelligence and Virtual Reality Applications. He is also associated with the Future Workspaces Research Centre at the University of Salford where he has previously worked for a number of years as a Senior Research Scientist at its Virtual Reality Centre.