Abstract—This paper presents an adaptive sliding window flow control protocol for MPLS networks, based on estimating the available link bandwidth using Kalman Filtering enhanced by Bias Estimation. An optimal control algorithm is then implemented that minimizes the variance of queue length deviations from the desired target. The simulation results show that, with bias estimation, the bandwidth estimate converges much faster than with ordinary Kalman filtering. We also achieve the goal of maximizing the bandwidth link utilization efficiency while minimizing the packet loss rate.

Keywords: Sliding Window, MPLS, Enhanced Kalman Filtering, Bias Estimator, Flow Control

I. INTRODUCTION

Multiprotocol Label Switching network or MPLS is a data carrying mechanism in packet switched networks [1]. When a network is designated to be in an MPLS domain, routers located at the edge of the domain, called Label Edge Routers or LERs, establish connection paths with other routers in the MPLS domain. Upon connection, packets entering the ingress LER are affixed with a label generated by the Label Distribution Protocol (LDP) [2]. Labeling thus calls for all routers inside the MPLS domain to route through labels instead of a lookup in the IP routing table.

The manner of transmitting and receiving packets in an MPLS network is similar to that inherent in a TCP/IP network in that a flow control mechanism is implemented in an effort to prevent a transmitting party from sending information at a greater rate than the capacity of the receiving party [3]. The Sliding Window mechanism allows more than one packet to be transmitted at any one time. This mechanism greatly improves the link efficiency of transmission over the Stop and Wait flow control protocol. The window size at a discrete time $k$ or $W(k)$ is normally fixed at a certain value. This can lead to traffic congestion and packet losses due to buffer overflow. Such would be the case for transmission of a real-time video conferencing session, where such losses would result in a low Quality of Service (QoS).

Many adaptive sliding window mechanisms have been proposed over the years, most of which are intended for congestion control. Raj Jain [4] proposed an adaptive mechanism using a timeout as an indicator of congestion. Mehdi Nesseh [5] proposed the mechanism based on transmission errors. Christensen [6] proposed a ‘Jump policy’ which adjusts the window size according to the packet loss and the window size at which the loss occurred. Phromsuphorn [7] proposed an algorithm based on the prediction of the distance between the transmitting and the receiving parties, and then using this information to adjust the window size.

In this paper, we propose an adaptive sliding window flow control protocol based on predicting the traffic rate passing through a router using Kalman Filtering which then provides an estimate of the available link bandwidth. The ordinary Kalman Filter is enhanced with a bias estimator. Optimal Minimum Variance Window Control [8] is then performed based on propagation delay information and the estimated link bandwidth. By this means, we optimize the utilization of the available link bandwidth as well as throughput in real time.

The present paper is organized as follows: Section II explains the flow control concepts in general, Enhanced Kalman Filtering, and Optimal Minimum Variance Window Control. Section III describes the simulation model of the MPLS network for the work investigated in this paper. Section IV presents performance behavior results and their analyses. Finally, conclusions are drawn in Section V.

II. OVERVIEW OF FLOW CONTROL MECHANISMS

A. Flow Control

Flow control is a technique for assuring that a transmitting party does not overwhelm the receiving party with data information [3]. Normally, the receiving entity has a limited amount of buffer storage for a transfer. Thus, in the absence of flow control, the buffer may fill up and overflow while old data is being processed. Stop-and-Wait protocol is the simplest flow control mechanism, in which the transmitting party is allowed to send only one packet at a time. Upon transmitting, it must then wait for a response known as Acknowledgement from the receiver. If the transmitting party is exceedingly far away from the receiving party (normalized round-trip time $\alpha > 1$), then the resulting link utilization will be low.

Greater link efficiency may be obtained by allowing multiple packets to be transmitted at a time. This technique is called Sliding Window Flow Control. The transmission link is treated as a pipeline that may be filled with frames in transit [9].
B. Enhanced Kalman Filtering

An ordinary Kalman Filter estimates the current state of the system as defined in (1), employing feedback of the noisy measurement (2). There are two types of update equations, namely time update and measurement update equations. The time update equations estimate the state of the system as well as the error covariance for prediction of future events. The measurement update equations correct the estimates in the form of feedback responses [10].

The system is defined by:
\[ x_k = Ax_{k-1} + B w_k \]
\[ z_k = H x_k + v_k. \]

The time update equations are:
\[ \hat{x}_k^- = Ax_{k-1} + B w_k \]
\[ P_k^- = AP_{k-1}A^T + B Q B^T. \]

The measurement update equations are:
\[ K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \]
\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-). \]

where \( \hat{x}_k^- \) is the a priori estimate of the system at time \( k \); \( P_k^- \) is the error covariance of the state estimate \( \hat{x}_k \); \( Q \) is the covariance of process noise \( w_k \); \( R \) is the covariance of the measurement noise \( v_k \) and \( K_k \) denotes Kalman Gain.

In light of sudden changes in the bias term, the ordinary Kalman Filter is no longer sufficient. Hence the bias term \( \beta(k) \) is modeled as follows [15]:
\[ \beta(k) = \beta(k) + w_k. \]

It is possible [17] to derive the optimum bias estimate through:
\[ \hat{\beta}(k) = (I - G_k S_k) \hat{\beta}(k-1) + G_k (z_k - H \hat{x}_k). \]

With:
\[ G_k = M_{s_{k-1}} (C_{s_k} V_k + B_k)^T R^{-1} \]
\[ S_k = C_{s_k} U_k + V_k \]
\[ V_k = U_k - K_s S_k. \]

The matrices \( U_k \), \( V_k \) and \( M_{s_{k-1}} \) are determined recursively, using:
\[ U_{s_{k-1}} = AV_k + B_k \]

The corrected state estimate for \( x_k \) is then given by:
\[ \hat{x}_k = \hat{x}_k^- + V_k \beta(k) \]

C. Optimum Variance Window Control

Our aim is to seek a window control law that achieves proportional fairness in the long run, obtaining maximum attainable link bandwidth and minimizing the probability of having buffer overflow. The latter two goals are the focus of the investigation described in this paper. From the result of [11], the first and the third goals may be achieved by designing the control to keep the long-run mean queue length of the bottleneck router at some desired level \( q_{ref} \), whilst minimizing the variance of the queue length in order to minimize the probability of packet losses through buffer overflow.

We proceed to find the window control law that minimizes the variance of the queue, bearing in mind that the law must also be compatible with the actual available information at the instant the optimal window \( W(k) \) is calculated [12].

The deviation of \( \Delta Q(k + 2) \) from its reference level is given by:
\[ \Delta Q(k + 2) = q_{ref} - W(k) - RTT_{base} \cdot B(k+1) \]

Where \( RTT_{base} \) denotes the round trip time and \( B(k) \) denotes the bandwidth attainable at instant \( k \). The sender’s most recent bandwidth measurement may be taken as:
\[ B(k-1) = W(k-1)/RTT_{base} \]

We may rewrite (15) in terms of the estimator bandwidth \( \hat{B}(k+1|k-1) \) based on the available information up to instant \( k-1 \):
\[ \Delta Q(k + 2) = q_{ref} - W(k) - RTT_{base} \cdot [\hat{B}(k+1|k-1) + \hat{B}(k+1|k-1)] \]

Where \( \hat{B}(k+1|k-1) \) denotes the error in bandwidth estimation

The variance in the queue length may now be written, from (17), as:
\[ \text{var}[\Delta Q(k + 2)] = [q_{ref} - W(k) + RTT_{base} \cdot \hat{B}(k+1|k-1)]^2 + \text{var}[RTT_{base} \cdot \hat{B}(k+1|k-1)] \]

It is evident from (18) that the second term is not dependent on \( W(k) \). For minimum variance control of queue, we therefore require that the first term vanishes. Hence:
\[ W(k) = q_{ref} + RTT_{base} \cdot \hat{B}(k+1|k-1) \]
This is the optimal variance control law. It may be given a simple interpretation that the optimal window size is the sum of the desired number of packets in the buffer plus the product of the round trip time and the two-step ahead prediction of available bandwidth [11]. Finally, the optimal sending rate is:

\[ R(k) = W(k)/\text{RTT}. \]  

III. PROPOSED PROTOCOL

A. System Description

The basic framework of our Adaptive Sliding Window Flow Control algorithm can be thought of as a system block diagram shown below.

![System Block Diagram](image)

Fig. 1. Framework of the Adaptive Sliding Window Flow Control algorithm

The traffic \( s(k) \) arriving at the input to the system in Fig. 1 can be envisaged as comprising two types: one type, denoted by \( \kappa \), is controllable. Its behavior is such that it is capable of adjusting its rate to the available bandwidth. The other type of traffic is denoted by \( \epsilon \), and is bursty, unpredictable, and presumably non-linear. Being real-time, it has a higher priority than the controllable (elastic) traffic \( \kappa \) [14].

The traffic predictor block as shown in Figure 1 lets the router compute an expectation of the traffic \( E(s(k)) \), and is in accordance with the Minimum Variance Window Control algorithm, whose function ensures the variance of the deviation of the queue length \( q(k) \) from a desired queue length \( q_{ref} \) is minimal statistically. The algorithm then adjusts the transmission window size \( W(k) \) according to (23) and the router’s transmission rate according to (20). The estimated bandwidth in (14) can then be used as \textit{a priori} knowledge for the traffic predictor in the next time step.

B. Modeling

The first step towards developing an approach to adaptive sliding window protocol would be to analyze the traffic of packets arriving into the Label Edge Router. To this end, we need a \textit{a priori} information concerning the state of the traffic \( s(0) \) before prediction. This information is derived from the Bandwidth measurement.

We postulate an auto-regressive (AR) model for the Bandwidth by the following equation at instant \( k \) [11]:

\[ B(k) = a_\omega B(k-1) + (1-a_\omega) \bar{B} + b_\omega w(k). \]  

The two-step ahead prediction is obtained as:

\[ \hat{B}(k+1|k-1) = a^2\bar{B}(k-1) + (1-a^2)\bar{B}(k-1). \]  

Hence, the window length is given by:

\[ W(k) = R_\kappa a_\omega^2 \bar{B}(k) + R_\epsilon (1-a_\omega^2) \bar{B}(k) + q_{ref}. \]  

We wish to compute the estimated bandwidth \( \hat{B}(k+1) \). Replacing \( k \) by \( k+1 \) in (12) and rewriting gives:

\[ \hat{B}(k+1) = a_\omega \hat{B}(k) + (1-a_\omega) \bar{B} + b_\omega w(k). \]  

In the formulation of Enhanced Kalman Filtering, the second term in the right-hand side of (24) is initially ignored because it is a bias term that is estimated separately. Hence, the state equation matrices in (1) and (2) are:

\[ A = a_\omega \]
\[ B = b_\omega \]
\[ H = 1. \]

The ordinary Kalman Filter time update equations are:

\[ \hat{s}(k) = A\hat{s}(k-1) \]
\[ \hat{P}(k) = \hat{P}(k-1) + BQ(k-1) \]

The measurement update expression is:

\[ \hat{x}(k) = \hat{x}(k) + K(k)\{y(k) - H\hat{x}(k)\}. \]

The Kalman Gain relation is:

\[ K(k) = \frac{P(k)H}{P(k)H^2 + R(k)}. \]

The set of ordinary Kalman Filter equations (27) – (30) being applied to estimate the state of the Bandwidth \( B(k) \) in (21) and (23) are only a first approximation of state estimation of the system, assuming that the Bias term \( \bar{B} \) is constant. However, in MPLS Networks, sudden changes in the Bias term exist, i.e. the mean of the traffic load changes.

The modeling of the bias term is, from (7),

\[ \bar{B}(k) = \bar{B}(k) + w. \]

Using (8) – (14), the corrected Bandwidth estimate \( \hat{B}(k) \) can be derived as:

\[ \hat{B}(k) = B(k) + V \bar{B}(k). \]
The Optimal Window Control Law (19) may now be implemented.

IV. PERFORMANCE RESULTS AND ANALYSES

A. Simulation over a link

We first study the performance of the Enhanced Kalman Filtering, and simulate the model described in Section III over a bottlenecked connection link. The diagram of such a link is depicted as indicated:

![Link Diagram](image)

Fig. 2. A Bottlenecked 8 Mbps link

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>PARAMETERS OF TRAFFIC MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Lossy Link (8Mbps)</td>
</tr>
<tr>
<td>Packet size (Kbits)</td>
<td>10</td>
</tr>
<tr>
<td>Arrival rate (packet/s)</td>
<td>800</td>
</tr>
<tr>
<td>Process Rate (μ) (packet/s)</td>
<td>1200</td>
</tr>
<tr>
<td>Time Delay (τ) (s)</td>
<td>0.005</td>
</tr>
</tbody>
</table>

The amount of traffic over an Ingress LER, obtained by simulating the model described in Section III, is shown in Figure 3.

![Traffic Graph](image)

Fig. 3. Traffic passing through an Ingress LER

The amount of traffic over an Ingress LER, obtained by simulating the model described in Section III, is shown in Figure 3.

The amount of packet loss is defined as the number of packets that are not forwarded to the next router after time delay $\tau$ has elapsed. The amount of packet losses incurred when Kalman Filtering is not in place is compared to that when Kalman Filtering is in place as shown in Fig. 4. It is seen that packet losses decrease by several orders of magnitude when the Kalman traffic predictor is in place.

![Packet Losses Graph](image)

Fig. 4. Packet Losses. Top. Without Kalman Estimation. Bottom. With Kalman Estimation

We now simulate the response of the Optimum Minimum Variance Window control, having obtained the traffic information from Kalman predictor. We set the queue reference length to be 50. The value of parameter $R_B$ was fixed at 0.1. The Bandwidth model autoregressive parameter $\alpha_{ar}$ was fixed at 0.9. Figure 5 shows the Queue Length of the buffer and Adaptive Sliding Window sizes. Intuitively, a low amount of queue in the buffer implies a low amount of packet loss. Nonetheless, this would also mean a low value of link utilization percentage. Our model ensures that the actual queue length deviates from the desired queue length with the minimum possible variance. This trades off packet loss against high link utilization, as will be shown.

![Queue Length Graph](image)

Fig. 5. Top. Window Size. Bottom. Buffer Queue Length

The performance of the Adaptive Sliding Window protocol by means of throughput measurements as well as the Link Utilization is shown in Fig. 6. The figure shows that the maximum attainable link bandwidth is 300 packets/sec. The Link utilization percentage is at 85% for a very considerable portion of our simulation time, with an average of 78%. It is shown in Fig. 6a that the Bandwidth estimates performance.
for the Bias Estimator converges faster and yields a more consistent estimate than the pure Kalman Filtering.

Fig. 6. Performance Measures of the Adaptive Sliding Window in MPLS Networks. Left. Bandwidth Performance. Right. Percentage Link Utilization

The same simulation was performed with $a_w = 0.95$, with every other parameters in Table I kept constant.

Fig. 7. Performance Measures of the Optimal Adaptive Window protocol in MPLS Networks with $a_w = 0.95$. Top. Bandwidth Performance. Bottom. Percentage Path Utilization

It is seen that the Link utilization percentage is increased by up to 15% from the case where $a_w$ is 0.90. We note that increasing the parameter $a_w$ makes a significant difference to the link utilization performance of the system.

B. Simulation over a small MPLS Network

We simulate the performance behavior of the following MPLS network whose topology is shown in Fig. 8. The parameters used for this simulation are given in Table I.

Fig. 8. MPLS Network Topology

We route two types of traffic—data and voice. Each type of traffic is designated the same Forward Equivalent Class (FEC). The MPLS domain is configured such that the Label Switch Path of the data traffic is as shown in Fig. 8. The voice traffic is load balanced. Sixty percent of voice traffic is routed to the Egress LER via LSR 2, while the rest of the voice traffic is routed to the Egress LER via LSR 3. Fig. 9 depicts the traffic information. Only traffic as predicted by Kalman Filter is shown.

Fig. 9. Traffic Information. Top Left. Data Traffic from the Ingress LER to LSR 1 and from LSR 1 to LSR 3. Top Right. Voice Traffic from Ingress LER to LSR 2. Bottom. Voice Traffic from Ingress LER to LSR 4.

Figure 10 shows Adaptive Sliding Window sizes. It is seen that Optimal Window Control automatically adjusts the window size to accommodate the traffic, based on the traffic Kalman Filter prediction.

Fig. 10. Sliding Window Information. Left. Data Traffic Sliding Window. Right. Voice Traffic Sliding Window.

Figure 11 shows the throughput performance as well as the link utilization performance.
Although the bandwidth performance is about the same as before around the non-zero value of traffic, the overall performance now undergoes degradation. The link utilization average degrades from 80% in the one link case to around 70% in the Network case as seen from Fig 9b) and Fig. 9c). Degradation results from the virtue of MPLS network, which requires all traffic of the same FEC, be routed to the same LSP. It is seen that the advent of load balancing also adds to the degradation effect in Link Utilization performance. It is seen that as more diversity in traffic class is introduced, the resulting traffic individual link resembles bursty traffic. This again proves our conjecture in using Kalman Filtering to predict the traffic.

V. CONCLUSION

We have presented a new Adaptive Sliding Window Flow Control mechanism which is applicable to MPLS networks. The mechanism requires the router to predict the traffic information via Kalman Filtering, enhanced by bias estimation, and then uses the information to implement Optimal Window Control, which maximizes the attainable bandwidth as well as the link utilization. We first showed through simulation of a single-connection system that traffic prediction by Kalman Filtering enhanced by bias estimation improves the performance of the Window Control.

We next validated our mechanism in a small MPLS network carrying two classes of traffic. The Bandwidth performance as well as the link utilization performance degrades when load balancing is in force. Discovering a mechanism to overcome such effect in order to maintain high link utilization performance as provided in the single-connection case is left for future work. Other areas of future research would include investigating the effect of link failure on the Adaptive Sliding Window mechanism, and how to counteract such effects.

REFERENCES