

# Network State Estimation Using Smart Experts

Yalda Edalat

Department of Computer  
Engineering  
University of California, Santa Cruz  
Santa Cruz, USA  
Email: yalda@soe.ucsc.edu

Jong-Suk Ahn

Department of Computer  
Engineering  
Dongguk University  
Seoul, Korea  
Email: jahn@dgu.edu

Katia Obraczka

Department of Computer  
Engineering  
University of California, Santa Cruz  
Santa Cruz, USA  
Email: katia@soe.ucsc.edu

## ABSTRACT

Several core network protocols and applications adjust their operation dynamically based on current network conditions. TCP and IEEE 802.11 are notable examples, both of which periodically adapt the retransmission timeout and the contention window size depending on the average round trip time and the degree of collisions, respectively. Consequently, accurate network state estimation is critical to the performance of networks and their applications.

In this paper, we present a novel mechanism to estimate "near-future" network state based on past network conditions. Smart Experts for Network State Estimation, or SENSE, uses a simple, yet effective algorithm combining a machine-learning method known as Fixed-Share Experts and Exponentially Weighted Moving Average (EWMA). SENSE introduces novel techniques that improve the performance of the basic Fixed-Share Experts framework by: (1) making SENSE's accuracy considerably less sensitive to the number of experts; and (2) making SENSE more responsive to network dynamics at different time scales, i.e., long- and medium-term fluctuations as well as short-lived variations. We evaluate SENSE using synthetic and real datasets. Our results show that it yields superior performance for all datasets we used in our experiments when compared to "pure" Fixed-Share Experts and EWMA. We confirm that the performance of EWMA is quite sensitive to its "smoothing" factor, which specifies how much weight will be placed on the "past" versus the "present" when predicting the "future". Another key advantage of SENSE is that, unlike Fixed-Share Experts, it needs no a-priori information about the dataset. In our experiments, SENSE yields up to 24% and 30% prediction accuracy improvement over the Fixed-Share algorithm and EWMA, respectively.

## Categories and Subject Descriptors

C.2.5 [Local and Wide-Area Networks]: Internet

## General Terms

Algorithms, Measurement, Performance, Experimentation

## Keywords

Machine learning; network state estimation; expert framework; smart experts

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIQUITOUS 2014, December 02-05, London, Great Britain  
Copyright © 2014 ICST 978-1-63190-039-6  
DOI 10.4108/icst.mobiquitous.2014.257949

## 1. INTRODUCTION

Most computer network protocols and algorithms try to adapt to current network state via a number of operational parameters that dynamically estimate current conditions in the network. Notable examples include the Transmission Control Protocol (TCP) and IEEE 802.11 (Wi-Fi) which adjust the retransmission timeout and contention window size, respectively, according to network congestion and wireless channel state. More specifically, to recover lost packets in a timely manner yet minimizing the number of unnecessary retransmissions, TCP periodically evaluates the degree of network congestion under the assumption that network conditions will stay almost the same until the next evaluation period. It uses the round-trip time (RTT), i.e., the time between sending a segment and receiving confirmation from the other end that the segment was received, as a way to gauge network load. TCP adjusts its retransmission timeout, i.e., the interval of time the TCP sender will wait for a segment's acknowledgment from the TCP receiver before retransmitting the segment, based on TCP's current estimate of the RTT. To compute its estimate of the RTT, TCP runs a simple mechanism known as Exponentially Weighted Moving Average (EWMA) with one tunable parameter, which determines the relative weight between the current RTT measurement and the previous RTT estimate.

The IEEE 802.11 responds to congestion build up in the network by exponentially inflating its back-off window, which stipulates the average amount of time that a node should wait to transmit after a collision has occurred. The rationale for this exponential back off is that collisions are used as congestion indicators; and, after a failed attempt to transmit due to a collision, the transmitter needs to wait longer before trying again. To estimate the "near-future" channel state, IEEE 802.11 counts the number of consecutive collisions that took place during the current estimation time window and exponentially expands the size of the back-off window according to this collision count.

Clearly, the performance of these widely used network protocols heavily relies on how correctly their prediction mechanisms forecast "near-future" network state. Their implicit assumption is that network conditions change smoothly, i.e., that "near future" state is closely correlated to previous history. As a result, their performance can be negatively affected when their operational parameters are set without accurately accounting for network dynamics. TCP, for instance, statically presets the weight factor in its RTT EWMA equation irrespective of the target network environment and conditions. As previously pointed out, the fixed weight factor in TCP's RTT EWMA calculation is a relative ratio deciding how much the current RTT measurement and the current RTT estimate should influence the new RTT estimate. The more dynamic the network conditions, the more weight should be placed on the current RTT measurement. Therefore, to achieve better performance, the fixed

weight factor should change dynamically depending on network conditions.

IEEE 802.11 rigidly cold-starts and counts collisions at every new frame's transmission without considering previous channel state. This means that considerable resources may be wasted in the process of reaching an adequate congestion window since 802.11's network estimation technique does not keep track of the network state after successful transmissions.

Motivated by the need to accurately estimate near-term future network state that may slowly or rapidly change, this paper introduces Smart Experts for Network State Estimation (SENSE). SENSE is a simple, yet efficient machine-learning based predictor that is based on the Fixed-Share Experts approach [1] [2] [5] [6]. Unlike conventional network state estimators, SENSE provides a general framework that can incorporate any traditional estimator as an "expert" and then dynamically selects the best experts among the set of all experts being used depending on their performance. It chooses experts that more faithfully capture network dynamics by penalizing poorly performing experts.

The original Fixed-Share Experts algorithm [1], however, has four main drawbacks. First, a fix value within the range we are trying to predict is assigned to each expert at the beginning of each trial. Thus, the range of the estimation is required for proper assignment of these values. Second, its accuracy is sensitive to the number of experts and typically, the more experts, the more accurate the prediction since the algorithm basically singles out a few well-behaved experts among the set of competing experts. There is clearly, a "diminishing return" effect after the number of experts gets too high. Third, the "loss function" penalizing experts, relies exclusively on the magnitude of the current error, instead of whether errors have recently increased or decreased. Additionally, all poorly performing experts are equally penalized. Depending on the recent error variation history, the loss function should intensify or alleviate the penalty for each individual expert to accelerate convergence. Finally, since the original Fixed-Share algorithm is not equipped with a mechanism to offset what it has learned, it cannot swiftly adapt to abrupt changes even when recent measurements become distinctly different from previous ones.

To address these problems, SENSE introduces three techniques, namely: (1) smart experts, (2) META-learning, and (3) Level-Shift. To reduce the sensitivity to experts and eliminate the need for pre-trial data knowledge, SENSE employs EWMA equations with different weights as its experts and normalizes errors by the maximum observable output. SENSE's META-learning algorithm expedites convergence by tracing recent past history and adjusting each expert's penalty accordingly. Finally, the Level-Shift mechanism [11] employed by SENSE improves its response to sudden data changes by bounding SENSE's learning time window, and upon detecting dissimilar data patterns, SENSE reinitializes its tunable parameters and starts to relearn.

We evaluate SENSE using synthetic and real data. In all cases, SENSE outperforms predictors based on pure EWMA as well as Fixed-Share Experts. Furthermore, a key advantage of SENSE is that it automatically adjusts to the data it is trying to predict. As a result, SENSE yields superior performance for all datasets we used in our experiments when compared to "pure" Fixed-Share Experts and EWMA. Our results also indicate that the performance of EWMA is quite sensitive to its "smoothing" factor, which determines how much weight will be placed on the "past" versus the "present" when predicting the "future". Another key advantage of SENSE's ability to automatically adjust to the data is that, unlike Fixed-Share Experts, it

needs no a-priori information about the dataset and is minimally sensitive to the number of experts. In our experiments, SENSE yields up to 24% and 30% prediction accuracy improvement over the Fixed-Share algorithm and EWMA, respectively.

The rest of the paper is organized as follows. Section 2 presents some background on EWMA and the Fixed-Share Experts algorithm while section 3 describes SENSE in detail. Section 4 compares the performance of SENSE against EWMA and fix Fixed-Share Experts. Section 5 provides an overview of related work and finally, section 6 concludes the paper with directions for future work.

## 2. BACKGROUND

SENSE's predictor is based on a combination of the Fixed-Share Experts algorithm and EWMA. Therefore, as background information, in this section we review EWMA and the Fixed-Share Experts algorithms in details.

### 2.1 EWMA

EWMA based predictors, calculate an exponentially weighted mean of the previous data. Equation (1) shows the basic equation of exponential smoothing given by Hunter [13] where  $y_t$  and  $x_t$  represent, respectively, a sequence of data point that have been observed and sequence of output forecast by the predictor. Furthermore,  $\alpha$  in (1) is the "smoothing factor", a value between 0 and 1 specifying how much relative weight is given to previous estimates (i.e., the "past") versus new samples (the "present").

There has been no generally accepted statistical technique for choosing  $\alpha$ . Low values of  $\alpha$ , favor the "past" over the "present" when computing the current estimate, whereas with high  $\alpha$ , the "present" plays a more important role. In other words, low  $\alpha$  has EWMA acts as a low-pass filter smoothing out sudden fluctuations occurred in the input data series, while high  $\alpha$  acts as a high-pass filter hardly filtering out measurement noise.

The problem of using EWMA based predictors is choosing appropriate  $\alpha$  based on the dataset. SENSE runs a small number of EWMA's with different  $\alpha$ 's and, using the Fixed-Share Experts technique, dynamically picks the best performing EWMA depending on network dynamics.

$$x_t = \alpha * y_{t-1} + (1 - \alpha) * x_{t-1} \quad (1)$$

### 2.2 Fixed-Share Experts Algorithm

The Fixed-Share Experts algorithm is a member of the multiplicative weight algorithmic family that has shown to yield performance improvements in a variety of on-line problems [7]. Aiming at minimizing the prediction's error, this family of algorithms combines predictions of a set of experts  $\{x_1, x_2, \dots, x_N\}$  to compute the overall prediction denoted by  $\hat{y}_t$ . To denote the impact of each expert on the overall predictor, it associates each expert with a weight from  $\{w_1, w_2, \dots, w_N\}$ . After each trial, the weight of each expert is updated depending on the difference between its prediction and the real data represented by  $y_t$ . Weights of "well-performing" experts are not changed, while the weights of experts that are not performing well are reduced.

Several schemes have been proposed for updating experts' weights in multiplicative weight algorithms. Among them, Fixed-Share Experts [1] is well known due to its simplicity, and efficiency. The main idea of Fixed-Share Experts algorithm is to share a fixed fraction of the weight of "poorly-performing" experts among the other good experts.

Figure 1 shows the Fixed-Share Experts algorithm pseudo code with

$N$  experts applied to TCP's RTT estimation [2]. The *Prediction* step in Figure 1 computes the current prediction by summing, over  $N$  experts, the products of the expert multiplied by its current weight and then normalizing the result by the sum of the weights. Using a given "loss function", the *Loss function* step checks, at each prediction trial, how good of a prediction each expert yields. We have experimented with different loss functions and picked the one shown in Figure 1 for its efficiency as well as simplicity. In the *Exponential updates* step, the loss or error is then used to adjust the experts' weights. Finally, to prevent abrupt weight changes, the *Sharing weights* step redistributes evenly a certain fixed fraction of *pool*, which is the sum of a preset portion of each weight.

As an illustration, Figure 2 shows the implementation of the Fixed-Share Experts algorithm with  $N$  experts using a hardware block diagram. The shaded boxes on the left column and the middle column correspond, respectively, to the experts denoted as  $x_i$  and the penalty function. The process of updating weights and generating the final predictions are represented as a circuit employing the addition, division, and multiplication operators.

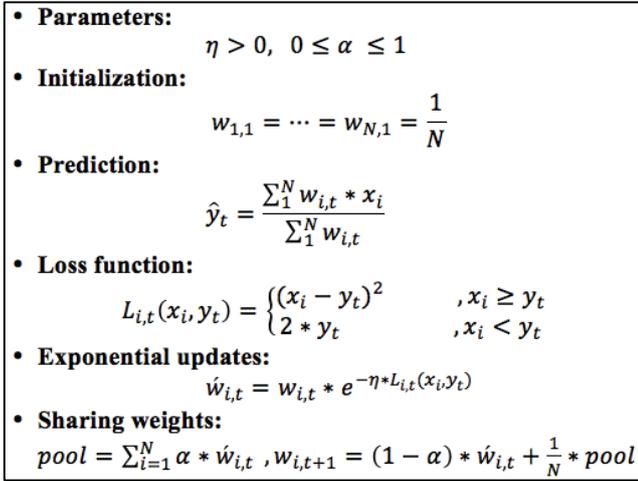


Figure 1: Fixed-Share Experts algorithm

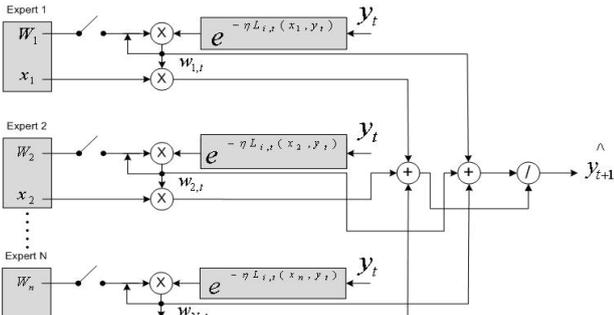


Figure 2: Hardware block diagram of Fixed-Share Experts algorithm

Equation (2) summarizes the pseudo code of Figure 1 into a mathematical formula. As shown in (2),  $\hat{y}_{t+1}$  can be represented by a sum of products of  $\alpha_{i,t}$  and  $x_{i,t}$  where  $\alpha_{i,t}$  is the experts' weights ( $0 < \alpha_{i,t} < 1$ ) which are dynamically and systematically adjusted and  $x_{i,t}$  is each expert's prediction. Equation (2) confirms that the Fixed-Share Experts algorithm is a selection process, which favors experts whose predictions are closer to the real data by incrementally

growing their weights, while reducing other experts' weights.

$$\hat{y}_{t+1} = \frac{\sum_{i=1}^N x_{i,t} * w_{i,t} * \exp\{-\eta * \sum_{i=1}^N L_{i,t}(x_i, y_t)\}}{\sum_{i=1}^N w_{i,t} * \exp\{-\eta * \sum_{i=1}^N L_{i,t}(x_i, y_t)\}} = \sum_{i=1}^N \alpha_{i,t} * x_{i,t}$$

$$\text{Where } \alpha_{i,t} = \frac{w_{i,t} * \exp\{-\eta * \sum_{i=1}^N L_{i,t}(x_i, y_t)\}}{\sum_{i=1}^N w_{i,t} * \exp\{-\eta * \sum_{i=1}^N L_{i,t}(x_i, y_t)\}} \quad (2)$$

Although the Fixed-Share Experts algorithm has been shown to perform well when estimating network variables, it still exhibits four main weaknesses. First, it must have a priori knowledge of the dataset's range in order to properly set the value of its experts. Second, the accuracy of the algorithm is quite sensitive to the number of experts whose values define the granularity over the range of values that the variable in question can assume. Third, since the loss function is statically predetermined regardless of the target environment and application, it is not always able to exhibit adequate convergence. Finally, Fixed-Share Experts is not sufficiently "agile" to adapt to rapid changes in the dataset since it "remembers" what it has learned and captures it into the weight of each expert. Even though weights of poorly performing experts decay over time, it is hard to swiftly adapt to rapid changes.

### 3. SENSE

This section provides a detailed description of our online estimator, SENSE, which employs a combination of Fixed-Share Experts with EWMA.

More specifically, SENSE is a modified version of the Fixed-Share Experts estimator, where, instead of fixed valued experts, EWMA filters are employed as experts. As shown in the *EWMA experts* step of Figure 3, the prediction of each expert,  $x_{i,t}$ , is calculated as a weighted sum of the previous seen data  $y_{t-1}$  and the previous prediction  $x_{i,t-1}$  where  $\alpha$  represents the relative weight between  $x_{i,t}$  and  $y_{t-1}$ . At the beginning of the whole process, each expert is assigned a weight,  $w_{i,1} = 1/N$ , where  $N$  is the total number of experts; each experts is also assigned an  $\alpha$  value between 0 and 1 which differentiates experts from each other.

As illustrated in the *Prediction* step of Figure 3, at every trial  $t$ , SENSE calculates the current prediction  $\hat{y}_t$  by adding the weighted predictions from  $N$  experts. After the completion of trial  $t$ , the *loss function* step in Figure 3 calculates the absolute difference between the actual outcome,  $y_{t-1}$ , and each expert's forecast  $x_{i,t}$  and then normalizes this error with the maximum observable outcome  $y_{max}$ . Finally, the *loss function*,  $L(x_{i,t}, t)_{i,t}$ , is set to either the normalized error  $NE_{i,t}$  or the *NULL function* depending on the size of errors. If  $NE_{i,t}$  lies within the satisfactory boundary  $EL$ , SENSE does not penalize experts differently than the original Fixed-Share Experts algorithm, which constantly adjusts the weight until the prediction equals the outcome. Here,  $EL$  can be set to any fraction between 0 and 1 according to the accuracy required by the application.

After that, SENSE runs *META-learning* step, which either multiplicatively increases or decreases  $\eta_{i,t}$  by  $\beta$  if three consecutive errors keep growing or shrinking respectively. Otherwise, it does not change  $\eta_{i,t}$ . To prevent each expert's  $\eta$  from being unrealistically too small or too large,  $\eta_{i,t}$ 's range is specified as  $[\eta_{min}, \eta_{max}]$ . We explore how  $\eta_{i,t}$ 's range impact SENSE's behavior in section 4.4. The goal of *META-learning* step is to speed up convergence of each expert's prediction to the observed outcome. For updating  $w_{i,t}$  with what has been learned, *Weight update* step multiplies  $w_{i,t}$  with

the exponent function whose power is a product of a  $L(x_{i,t})_{i,t}$  and learning factor  $\eta_{i,t}$ .

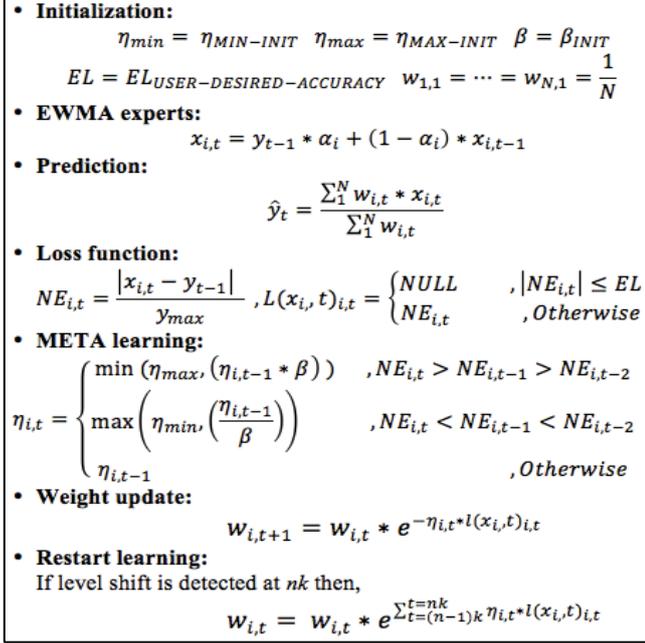


Figure 3: SENSE algorithm

Finally, SENSE employs a *Level-Shift* step [11] to detect any significant change in the mean of the observed data. Suppose  $\{X_1, X_2, \dots, X_n\}$  is the sequence of data, where  $X_1$  is the first data after the last detected level shift. The measurement  $X_k$  is an increasing (decreasing) level shift if it satisfies the following three conditions:

- 1) Data  $\{X_1, X_2, \dots, X_{k-1}\}$  are all lower (higher) than the data  $\{X_k, \dots, X_n\}$ ,
- 2) The median of  $\{X_1, X_2, \dots, X_{k-1}\}$  is lower (higher) than the median of  $\{X_k, \dots, X_n\}$  by more than a relative difference  $\chi$ , and
- 3)  $k + 2 \leq n$ .

The last condition assists to prevent misinterpreting an outlier as a level shift. Upon the detection of a level shift, SENSE restarts its experts by only considering data after the level shift occurrence and resetting  $\eta$  for each expert. This means that the weight of each expert is determined only by the accuracy of prediction after the last level shift. In other words, the Level-Shift step slides its learning window to include only data after the last level shift into the weight of each expert.

In summary, SENSE employs three main techniques as follows:

- *Smart experts* reduce the sensitivity to experts and eliminate the need for pre-trial data knowledge. SENSE employs EWMA equations with different weights as its experts and normalizes errors by the maximum observable output.
- *META-learning* expedites convergence by tracing recent past history and adjusting each expert's penalty accordingly.
- *Level-Shift* improves SENSE's response to sudden data changes by bounding SENSE's learning time window; upon detecting dissimilar data patterns, SENSE reinitializes its tunable parameters and starts to relearn.

## 4. EXPERIMENTS AND RESULTS

We experiment with SENSE, using a variety of input data and compare SENSE's performance against that of the original Fixed-Share Experts algorithm and EWMA.

In the first set of experiments, we evaluate SENSE on synthetic data that exhibit different periodic patterns. We use both sine and square wave signals with a range of frequencies. This experiment systematically tests how well SENSE can track the variation of input data over a wide spectrum of frequencies when compared to EWMA with different values of its smoothing factor,  $\alpha$ .

For a thorough comparative study, we also apply SENSE to the same set of RTT dataset used in [2] and compare the result against: (1) the original Fixed-Share Experts algorithm, (2) Jacobson's TCP RTT estimation algorithm [14] (which is a variant of EWMA), and (3) "pure" EWMA with different smoothing factors.

In addition, we run SENSE over real collision rate data collected from a production Wireless LAN environment where access points (APs) periodically collect traffic and load statistics such as the number of retransmissions, total number of frames transmitted, etc. Finally, we investigate the effect of SENSE's parameters on its overall accuracy.

### 4.1 Datasets with Periodic Patterns

These first sets of experiments compare SENSE's accuracy with EWMA when estimating datasets that follow periodic patterns. We use a dataset consisting of 1,000 samples. For the sine wave pattern, these samples create one period for 0.001 Hz and 200 periods for 0.5 Hz. The amplitude of our sine waves fluctuates between 0.25 and 0.75. For the square wave, these samples generate 40 periods for 0.025 Hz and 200 periods for 0.5 Hz. The amplitude of our square waves fluctuates between 0.1 and 0.7.

Choosing the best  $\alpha$  value depends on data autocorrelation and is a key factor for EWMA based estimators' performance. Values of  $\alpha$  closer to one have less of a smoothing effect and give more weight to recent changes in the data, while values of  $\alpha$  closer to zero have a greater smoothing effect and are less responsive to recent changes. We show that SENSE eliminates EWMA's dependency on  $\alpha$ . Note that the best value of  $\alpha$  needs to be decided based on the rate of changes in the data. Smaller  $\alpha$  worsens the accuracy for rapid changes, while larger  $\alpha$  degrades the accuracy when data fluctuations are smoother.

Our implementation of SENSE runs four EWMA experts with  $\alpha$  values uniformly distributed between 0 and 1, i.e., 0.2, 0.4, 0.6, and 0.8; we set  $\beta$  and  $EL$  to 2 and 0.01, respectively. In 5.4, we examine the impact of SENSE's parameters on SENSE's performance and show that SENSE's accuracy is quite insensitive to the number of experts and  $\beta$ . As previously discussed,  $EL$ 's value is set depending on the application's accuracy requirements. In order to cover low, medium, and high EWMA smoothing factors, we test EWMA with three  $\alpha$  values, namely: 0.25, 0.5, and 0.75. As for the input data function, we use two patterns: sine- (results plotted in Figure 4) and square waves (results shown in Figure 5).

Figure 4 plots the average error of SENSE and three EWMA filters as a function of the sine wave frequency. Each point in this figure is calculated by averaging the absolute error of all 1,000 samples. At higher frequencies, the input's current value tends to be further apart from the last outcome so that it is harder to accurately predict. This figure confirms that SENSE produces lower average error than any of the three EWMA filters over the entire frequency range. This is especially true at higher frequencies. As the frequency goes up,

errors from EWMA filters rise steeply regardless of the  $\alpha$  value. EWMA with higher  $\alpha$  tends to exhibit better accuracy over the lower frequency range, while EWMA with lower  $\alpha$  performs better for frequencies higher than 0.1 Hz. The reason for this phenomenon is that at low frequencies, where each sample is very similar to its previous sample, EWMA with higher  $\alpha$  (which places more weight on recent trials) outperforms EWMA with lower  $\alpha$  (that puts more weight on the history). When the frequency increases, recent trials are less correlated to the upcoming trial. Therefore, lower  $\alpha$  yields better accuracy.

In contrast to "pure" EWMA, SENSE dynamically adapts according to the frequency by choosing an appropriate EWMA expert for a given frequency range. As the frequency increase, SENSE shifts its reliance from EWMA with higher  $\alpha$  to EWMA with lower  $\alpha$ .

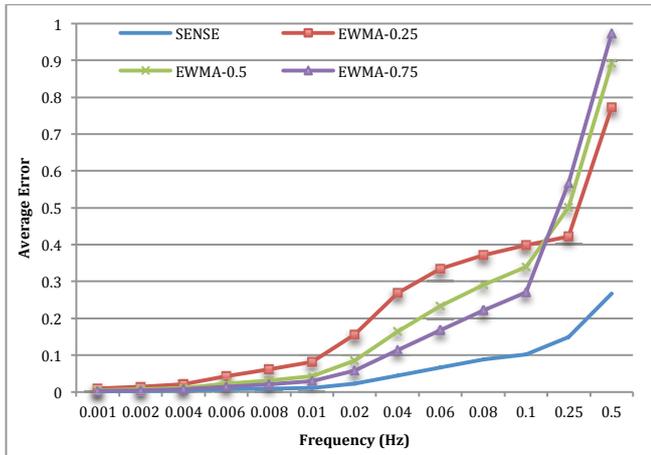


Figure 4: Average error comparison over sine waves

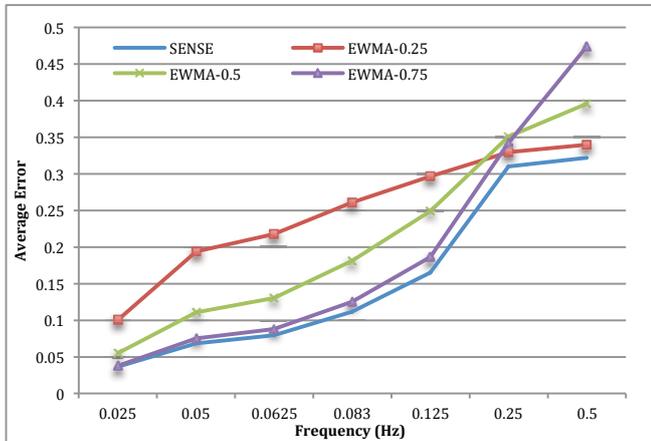


Figure 5: Average error comparison over square waves

Figure 5 shows the average error of SENSE and three EWMA filters when driven by square waves. This figure exhibits very similar trend as Figure 4 where SENSE outperforms all EWMA predictors at all frequencies. SENSE's smart experts are able to automatically switch between EWMA with high  $\alpha$  value at low frequencies and EWMA with low  $\alpha$  over the high frequency range.

## 4.2 Estimating TCP Round-Trip Times (RTT)

We also evaluated SENSE's accuracy when applied to real datasets. As discussed in section 1, TCP, one of the most widely deployed Internet protocols, uses round-trip time (RTT) as an indication of

network load. TCP employs its RTT estimates to trigger TCP's core functions such as error- and congestion control. Motivated by how critical accurate RTT estimates are for TCP's performance, we evaluate SENSE's accuracy in estimating RTTs in comparison to the Fixed-Share Experts algorithm employed in [2], as well as TCP's original RTT estimator based on Jacobson's well-known EWMA variant [14] as shown in Equation (4), where  $\alpha$  is typically set to 0.85.

$$Est\_RTT = \alpha * Est\_RTT + (1 - \alpha) * RTT \quad (4)$$

For these experiments, we use the RTT dataset in [2]. These RTTs were measured when a 16 MB file was transferred over a real network. As shown in Figure 6, SENSE is able to keep track of the RTT variations more faithfully than Fixed-Share Experts and Jacobson over the entire observation period. Table 1 summarizes the average normalized error of SENSE, the three different EWMA filters, Fixed-Share Experts and Jacobson's algorithms when applied to the same RTT data of Figure 6. In order to calculate the average normalized error, we first divide the absolute error of each sample by the real data it is trying to predict; then, we average these normalized errors. To compute the error ratio, we choose SENSE's average normalized error as baseline. Then, we calculate the other methods' relative error compared to SENSE as the difference between their average normalized error divided by SENSE's average normalized error. The resulting error ratio confirms that SENSE's accuracy outperforms both Fixed-Share Experts and EWMA.

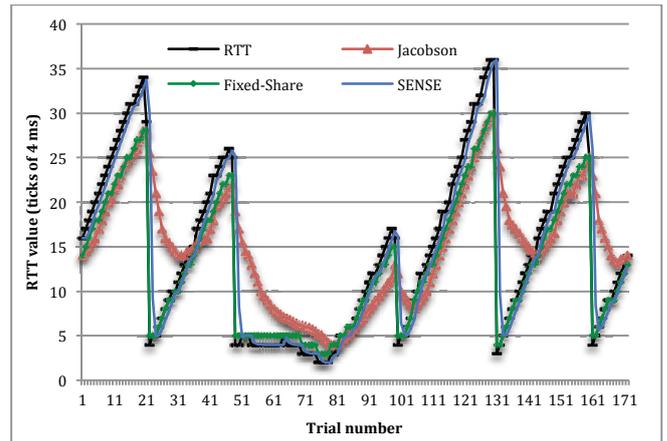


Figure 6: RTT prediction by SENSE, Fixed-Share Experts and Jacobson for each data sample (represented by a trial number)

Table 1: Average normalized error comparison for RTT

	SENSE	Fixed-Share	Jacobson	EWMA-0.25	EWMA-0.5	EWMA-0.75
Average normalized error	0.27	0.33	0.79	0.57	0.39	0.32
Error ratio (%)	-	24%	191%	111%	44%	20%

## 4.3 Estimating Collision Rates

To further evaluate SENSE's ability to forecast network dynamics in real environments, we applied SENSE to collision rate datasets measured in a production Wireless LAN (WLAN) environment. Collision rates were collected at access points (APs) as they send traffic to a node associated with it while other associated nodes

concurrently communicate with the AP, as they usually do. Specifically, we transmit 100 Mbps of UDP traffic from the AP to a node for 200 seconds while we simultaneously run different types of traffic between interfering APs and interfering nodes (i.e., located close to the node receiving data from the AP). Collision rates are calculated every second as the ratio of the number of retransmitted packets to the total number of transmitted packets. Since the test AP and the test node are physically close to one another, we assume that retransmitted packets are solely due to collision, and not to noise interference.

Figure 7 depicts how SENSE tracks a time series of real collision rates gathered from the test network for 200 seconds. We observe from Figure 7 that, initially, the dataset contains considerable "noise" caused by bursty traffic generated by short-lived flows from applications like the Web. After 100 trials (seconds), longer-lived flows resulting from traffic such as wireless video transmission becomes dominant, yielding "smoother" collision rate variations. Figure 7 shows while SENSE does not closely follow the sudden jumps in the first half of the time series, it is capable of accurately tracking the variations observed in the second half of the graph.

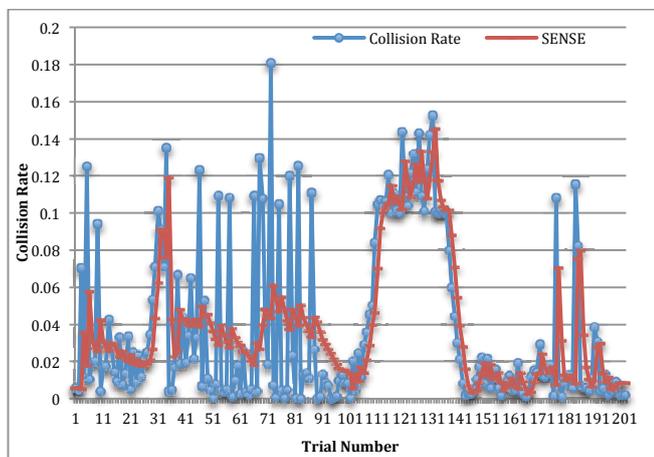


Figure 7: Trace of SENSE's collision rate prediction for each data sample (represented by a trial number)

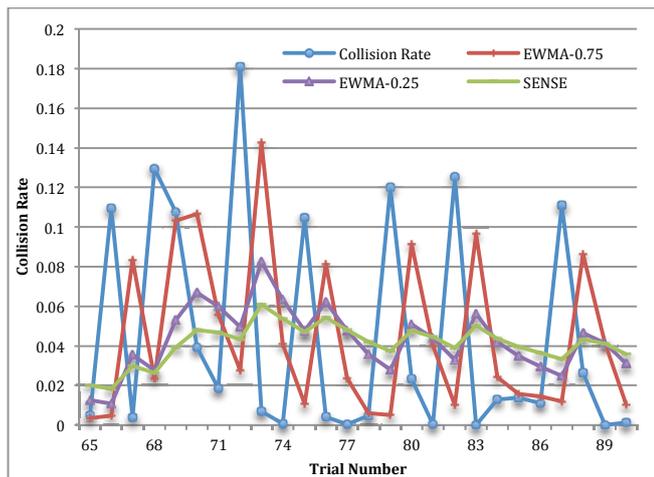


Figure 8: SENSE vs. EWMA for highly variant portion of collision rate

Figure 8 shows a closer view of the behavior of SENSE compared against two EWMA filters over a 25-second interval between 65-90

seconds of Figure 7. Note that in this span of time, data fluctuate significantly, which makes it very difficult for any predictor to predict accurately. In this period, SENSE behaves like a low-pass filter, e.g., EWMA with  $\alpha$  set to 0.25, while the curve corresponding to EWMA with  $\alpha$  value of 0.75 looks like the real data but delayed by a full trial, which results in the highest error. Table 2 summarizes the results shown in Figure 8 by comparing the average error and error ratio for the first 100 trials of the collision rate dataset. It confirms that, for the first half of the dataset which is quite "noisy" and, as a result, EWMA with  $\alpha = 0.25$  outperforms  $\alpha = 0.75$ , SENSE acts as an EWMA predictor with lower  $\alpha$  and shows slightly better performance than EWMA with  $\alpha = 0.25$ .

Figure 9 zooms in the performance of SENSE and two EWMA filters over the interval of 100-145 seconds in Figure 7. As shown in Figure 9, SENSE quickly catches up with collision rate changes and behaves similarly to EWMA with  $\alpha = 0.75$  (acting as a high-pass filter). In contrast, EWMA with  $\alpha$  value of 0.25 lags behind and cannot keep up with the collision rate variation. During this period, EWMA with  $\alpha$  value of 0.25 exhibits poor performance comparing to the other methods.

Table 2: Average Error for First 100 trials of Collision rate Data

	SENSE	EWMA-0.25	EWMA-0.5	EWMA-0.75
Average error	0.0323	0.0333	0.0337	0.0356
Error ratio (%)	-	3%	4%	10%

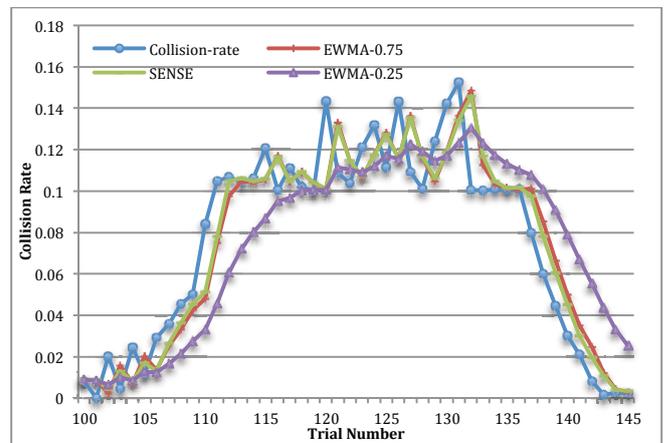


Figure 9: SENSE vs. EWMA for smooth portion of collision rate

Table 3 lists the average error and error ratio for the last 100 trials of Figure 7. During this interval where EWMA with  $\alpha = 0.75$  is clearly a better choice, SENSE behaves as EWMA predictor with high  $\alpha$  but achieves slightly higher accuracy.

Table 4 summarizes the average error and error ratio of the five different forecast schemes over the whole collision rate dataset depicted in Figure 7. It confirms SENSE's ability to automatically adapt its performance based on network dynamics. In the case of uncorrelated behavior, SENSE gives more weight to experts with low  $\alpha$  and in the case of correlated data, more weight is given to experts with high  $\alpha$  value. Since EWMA does not have this capability, for the first half of the dataset, EWMA with  $\alpha = 0.75$  is worse than SENSE by 10% (from Table 2) and for the second half of

the dataset, EWMA with  $\alpha = 0.25$  is significantly worse than SENSE (30% from Table 3). Table 4 clearly evidences that SENSE yields higher accuracy when compared to all the other four methods by at least 7% for the complete dataset. We also ran EWMA with weight matching our SENSE's highest and lowest  $\alpha$  values (0.8 and 0.2) and we saw no significant difference in EWMA average error results compared to the ones we have presented in Table 4. This comparison confirms SENSE's dynamic behavior to selectively and swiftly chooses the best expert according to the observed network dynamics. During noisy periods in the dataset, SENSE picks an expert with low  $\alpha$  while during periods when the data changes more smoothly, SENSE prefers an expert with high  $\alpha$  value.

**Table 3: Average Error for Second 100 trials of Collision rate Data**

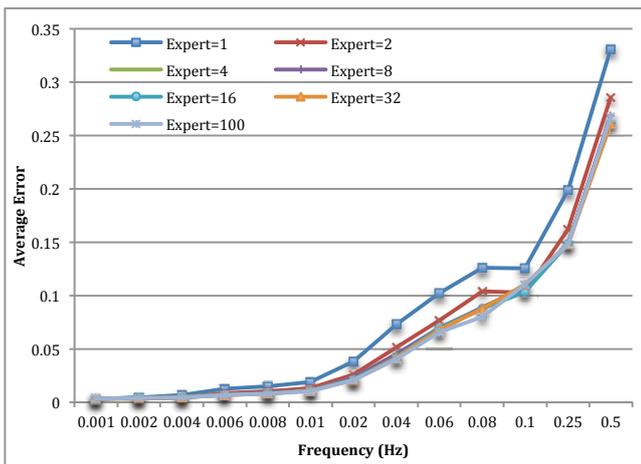
	SENSE	EWMA-0.25	EWMA-0.5	EWMA-0.75
Average error	0.0138	0.0180	0.0152	0.014
Error ratio (%)	-	30%	10%	3%

**Table 4: Average Error for collision rate dataset**

	SENSE	EWMA-0.25	EWMA-0.5	EWMA-0.75	Fixed-Share
Average error	0.0233	0.0258	0.0252	0.0249	0.0256
Error ratio (%)	-	11%	8%	7%	10%

#### 4.4 Impact of Parameters

In this section, we evaluate the effect of SENSE's tunable parameters such as number of experts,  $\beta$ ,  $\eta_{min}$  and  $\eta_{max}$ . Although results presented in this section are from experiments using datasets following sine wave patterns only, we observed similar results when we ran these experiments with our other datasets.

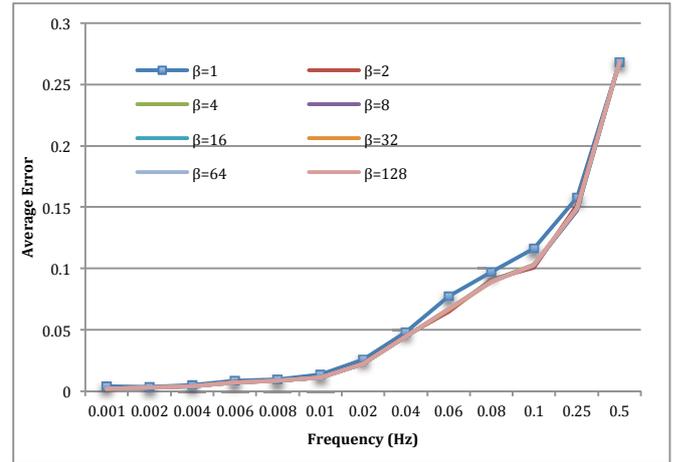


**Figure 10: SENSE's sensitivity to number of experts over sine waves**

Figure 10 displays SENSE's sensitivity to the number of experts and verifies that SENSE's performance barely changes when the number

of experts increases beyond 2. In our experiments, SENSE uses 4 experts with  $\alpha$  values of 0.2, 0.4, 0.6 and 0.8, which are uniformly distributed between 0 and 1.

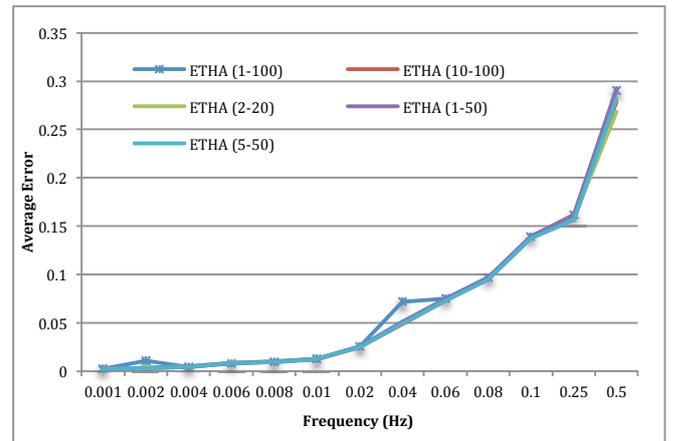
Figure 11 shows the impact of META-learning's  $\beta$  parameter on SENSE's behavior by plotting the average error-frequency curves for different  $\beta$  values. We observe that the difference in accuracy is almost indistinguishable for different  $\beta$ . This can be explained by the fact that each expert does its best to keep track of the input data. META-learning is invoked only when errors tend to continuously increase or decrease since it is designed to severely penalize static experts that maintain their prediction regardless of current measurements.



**Figure 11: SENSE's sensitivity to  $\beta$  over sine waves**

Figure 12 displays the impact of  $\eta_{min}$  and  $\eta_{max}$ , which are also META-learning parameters. The average error rate is shown in this figure for each pair of  $(\eta_{min}, \eta_{max})$  applied to sine waves with different frequencies. As it is shown, different boundary limits do not have significant effect on SENSE's performance. In our experiments, we limit  $\eta$  value between 10 and 100.

Setting Level-Shift parameters depends on the application's accuracy requirements. By choosing small values for  $n$ ,  $k$  and  $\chi$ , level shift detection becomes more sensitive. In our experiments, we set  $n$  and  $k$  to 8 and 2 respectively. The value of  $\chi$  is proportional to the median of  $\{X_1, X_2, \dots, X_{k-1}\}$ . We set this ratio to 1/3 for our tests.



**Figure 12: SENSE's sensitivity to ETHA-min and ETHA-max over sine waves**

#### 4.5 Impact of Level-Shift and META-learning

We evaluate the effect of the Level-Shift and META-learning methods on SENSE's performance. Figures 13 and 14 show the increase in accuracy when SENSE uses: (1) Level-Shift only, (2) META-learning only, and (3) Combined Level-Shift and META-learning. Both figures confirm that these techniques improve the performance of SENSE. Note that the improvements resulting from Level-Shift on RTT are much higher than on collision rate. The reason is that the RTT data has a large number of level shifts and SENSE can detect them and adjust its experts to follow the variations in the data. On the other hand, in the collision rate dataset, data fluctuates significantly and does not trigger the Level-Shift mechanism.

Similarly to Level-Shift, META-learning yields larger contribution to SENSE's performance for the RTT dataset than collision rate. And again, the reason is that the RTT dataset exhibits smoother behavior; therefore, META-learning is able to effectively increase the weight of good experts and decrease the weight of bad experts, which improves SENSE's performance overall. Consequently, the combined improvement of both techniques for the RTT dataset is almost 25% and just below 10% for the collision rate dataset.

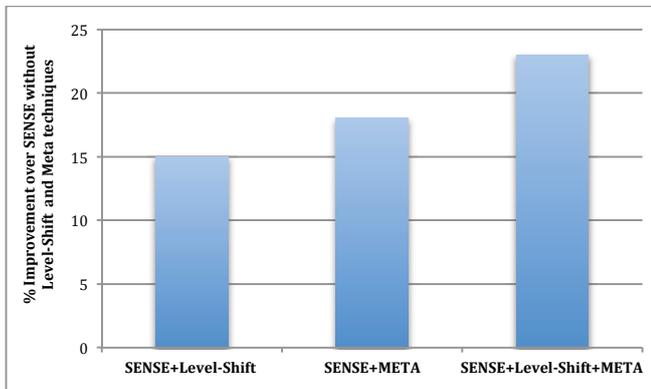


Figure 13: Impact of Level-Shift and META-learning methods on RTT dataset

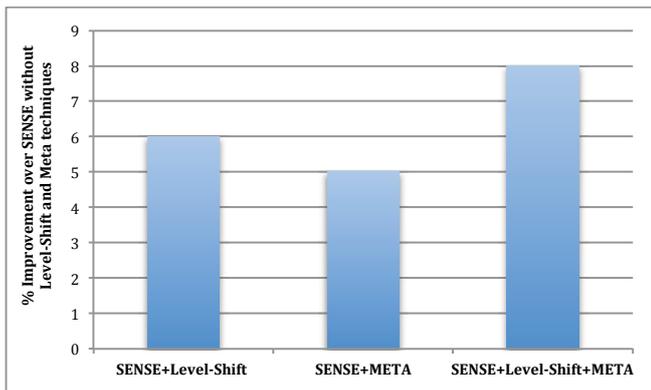


Figure 14: Impact of Level-Shift and META-learning methods on collision rate dataset

### 5. RELATED WORK

Several network protocols and applications make use of heuristics to estimate and adapt to the dynamics of the underlying network. Since the literature on the topic is quite extensive, in this section, we focus on reviewing work that is more closely related to ours.

EWMA is a well-known technique adopted by several communication protocols. As previously pointed out, TCP uses EWMA to estimate near-term round-trip time (RTT), which is used to set TCP's retransmission timeout (RTO). Since, depending on the network environment, RTTs may vary considerably in short timescales, a number of mechanisms have been proposed to either replace or augment EWMA. DualPats, a real time TCP throughput prediction service for distributed applications, was introduced in [12]. It utilizes EWMA to make throughput predictions of large transfers augmented with active probing. In [11], EWMA along with other simple linear predictors was employed to show that, in general, history-based methods predict the throughput of TCP transfers more accurately than formula-based techniques, i.e., mathematical models that express TCP performance as a function of network path characteristics.

More recently, a few efforts have used machine-learning techniques to estimate near-term network variables. For instance, the work in [2] proposed a TCP RTT predictor based on a simple yet efficient machine-learning algorithm called Fixed-Share Experts [1]. The results presented in [2] show that, for a variety of network scenarios and conditions, the proposed Fixed-Share Experts based predictor was able to improve RTT estimation significantly (thus yielding higher throughput) compared to existing approaches. Support Vector Regression (SVR) [9] also introduced a machine-learning method, which can accept multiple inputs to generate accurate predictions. This method was used in [10] to predict the end-to-end TCP throughput for arbitrary file sizes.

A variant of the Fixed-Share Experts approach has also been employed in the context of medium-access control (MAC). More specifically, in [8], a collision-free schedule based MAC that uses fixed-share experts to predict offered traffic load was proposed. Simulations as well as testbed results show the benefits of traffic prediction to schedule flows at the MAC layer in terms of delivery delay and delivery ration when compared to contention based MAC protocols. In [3], a method to predict direct and staggered collision probabilities of each node in WLANs has been introduced. Using information from an access point (AP) about network traffic broadcast as well as the AP's local measurements, each node obtains a spatial picture of the network in order to estimate probabilities of collisions locally. Similar techniques to the one used in [3] have been employed in [4] to improve throughput and link adaption in 802.11 networks with hidden terminals. In particular, a link adaption algorithm, in which nodes estimate the channel conditions by comparing the observed loss statistics to the expected loss statistics based on the estimated collision probability, is employed to select the ideal modulation rate under the estimated network conditions.

### 6. CONCLUSIONS AND FUTURE WORKS

In this paper, we introduced SENSE (Smart Experts for Network State Estimation) a novel network state predictor based on a simple, yet efficient machine-learning technique called Fixed-Share Experts. SENSE improves the Fixed-Share Experts algorithm by employing Exponentially Weighted Moving Average (EWMA)-based "smart" experts, META-learning, and Level-Shift techniques. Our experiments on both synthetic and real datasets confirm that SENSE can automatically adapt to fluctuations of different time scales, which sets it apart from "static" techniques such as "pure" EWMA and Fixed-Share Experts. Our experiments over a large set of data indicate that SENSE provides up to 24% and 30% prediction accuracy improvement over the Fixed-Share algorithm and EWMA, respectively.

As future research directions, we plan to apply SENSE to various network protocols such as IEEE 802.11e and X-MAC, which require channel state estimation to achieve better performance. To flexibly manage the degree of differentiation among classes of IEEE 802.11e traffic, we plan to adjust the protocol's contention window based on the collision rate forecast by SENSE. In power-aware MAC protocols such as X-MAC, we will develop an algorithm to dynamically adjust the sleep time of nodes according to the traffic load predicted by SENSE. We will also continue to improve SENSE; for example, we plan to devise a mechanism that allows the experts' smoothing factor,  $\alpha$ , to be automatically derived based on the input data.

## 7. ACKNOWLEDGMENT

This research has been partly supported by grant No. NRF-2012R1A1A2020051 from the Basic Science Research Program through the National Research Foundation (NRF) funded by Korea's Ministry of Education, Science and Technology and by grant CNS 1321151 from the US National Science Foundation.

## 8. REFERENCES

- [1] M. Herbster and M. Warmuth, "Tracking the best expert," in *Machine Learning*. Morgan Kaufmann, 1995, pp. 286–294.
- [2] B. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning framework for TCP round-trip time estimation". *EURASIP Journal on Wireless Communications and Networking*, March 2014.
- [3] M. Krishnan and A. Zakhor, "Local estimation of probabilities of direct and staggered collisions in 802.11 WLANs," in *Proc. IEEE International Workshops on Global Telecommunications (GLOBECOM)*, 2009, pp. 1 – 8.
- [4] M. Krishnan and A.Zakhor, "Throughput Improvement in 802.11 WLANs using Collision Probability Estimates in Link Adaptation," in *Proc. of IEEE Wireless Communications & Networking Conference*, Sydney Australia, April 2010.
- [5] N. Littlestone and M. Warmuth, "The weighted majority algorithm," in *Foundations of Computer Science*, 30th Annual Symposium on, oct-1 nov 1989, pp. 256 –261.
- [6] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R. Schapire, and M. Warmuth, "How to use expert advice," *Journal of the ACM*, Jan 1997, PP. 427-485.
- [7] D. Helmbold, D. Long, T. Sconyers, and B. Sherrod, "Adaptive Disk Spin- Down for Mobile Computers," *ACM/Baltzer Mobile Networks and Applications*, December 2000, 5(4):285–297.
- [8] P. Vladislav and K. Obraczka, "Collision-free medium access based on traffic forecasting," In *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2012 IEEE International Symposium on a IEEE 2012, pp. 1-9.
- [9] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression" *Statistics and Computing* , 2004, 14:199–222.
- [10] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to TCP throughput prediction." In *ACM SIGMETRICS Performance Evaluation Review*, ACM, 2007, vol. 35, no. 1, pp. 97-108.
- [11] He, Qi, Constantine Dovrolis, and Mostafa Ammar, "On the predictability of large transfer TCP throughput." In *ACM SIGCOMM Computer Communication Review*, 2005, vol. 35, no. 4, pp. 145-156. ACM.
- [12] D. Lu, Y. Qiao, P. Dinda, and F. Bustamante, "Characterizing and predicting TCP throughput on the wide area network," in *Proc. 25th IEEE ICDCS*, Columbus, OH, Jun. 2005, pp. 414–424.
- [13] J. S. Hunter, The Exponentially Weighted Moving Average, *Journal of Quality Technology* 18: 203-210, 1986
- [14] V. Jacobson, "Congestion avoidance and control," *SIG- COMM Comput. Commun. Rev.*, vol. 25, no. 1, pp. 157–187, 1995.