Efficient Link Management for the Wireless Communication of an Ocean Current Turbine Testbed

Ionut Cardei and Anthony Marcus
Department of Computer & Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL
{icardei, amarcu10} @fau.edu

Gabriel Alsenas
Southeast National Marine Renewable Energy Center
Florida Atlantic University
Boca Raton, FL
galsenas@fau.edu

***Abstract*—We describe a wireless link management framework for an ocean platform–to-shore communication system that uses time series forecasting to predict the available link capacity using ocean platform sensor data metrics to boost link robustness and to efficiently manage quality of service. Based on the predicted link capacity the OCTT Wireless Link (OWL) manager coordinates transmission scheduling of XML/HTTP sensor data at the web service layer and controls queue management for IP packet routing. To validate our framework, we developed a link management tool, the OCTT Wireless Link (OWL) manager to ensure optimal throughput and quality of service (QoS) for the wireless communication system linking ocean-based instrumented platforms with users on the shore. OWL applies sensor fusion to the platform attribute data which is then used to forecast the throughput of the wireless link in the harsh and rapidly fluctuating oceanic environment. OWL continually sends this forecast to the Queue Manager (QM) and when the signal power is forecast to drop outside the ideal range, using Linux networking tools, OWL provides bandwidth provisioning at the IP layer over each of the wireless radios in the network. This article describes our work on this project and experimentation with the OWL manager applied to sensory data collected from early stage OCTT platform testing.**

Keywords—Wireless Link Management; Quality of Service; Ocean to Shore Communication

I. INTRODUCTION

To help satisfy an ever growing global demand for energy, a team of engineers and scientists with the Southeast National Marine Renewable Energy Center (SNMREC) at Florida Atlantic University are preparing an ocean current turbine testbed (OCTT). In this article we describe the OCTT Wireless Link (OWL) manager that was developed to improve throughput and quality of service (QoS) for a wireless communication system linking the ocean instrumented platform with users on the shore. OWL applies sensor fusion to platform attribute data which is then used to forecast the state of the wireless link. The environment for an ocean-shore wireless link is harsh and rapidly changing, being mainly affected by antenna line of sight (LOS) obstruction, shadowing, rough surface reflection, and weather. OWL continually sends this forecast to another system component, the Queue Manager (QM), that controls link capacity allocation and transmission scheduling for web service streams carrying sensor data. Also, when the received signal power is forecast to drop, OWL adjusts in advance capacity allocation at the IP layer on the topside router for various packet flows (HTTP, SSH, RTP) sharing one or more wireless links, in order to prevent congestion, high delays for important data, and priority inversion.

Contemporary research has demonstrated that tides, waves, and ocean currents may be a viable source of renewable energy [1, 2, 3]. More recent analysis indicated that the Gulfstream current, located about 25 kilometers off the east coast of Florida, flows consistently at ~1.7m/s [4]. The OCTT will be anchored in the Gulfstream and is designed to evaluate and demonstrate prototype turbines for energy harvesting from these currents.

The OCTT prototype, described in [5], is equipped with a plethora of sensing components collecting a variety of turbine metrics such as power, vibrations, leaks, and pressure. Other sensors observe attributes of the surrounding environment and the physical state of the topside deployment platform. These sensor measurements are streamed to and stored by a Data Store subsystem on the topside platform. Onboard prognosis and health monitoring (PHM) programs access this data to monitor and forecast the health of the OCTT and the deployed turbine. Furthermore, authorized shore-side operators need access to collected data and must be able to send commands in real time via a wireless communication link. To ensure these components maintain adequate connectivity and reliability, we have developed the OWL manager.

In designing the wireless communication system for the OCTT, our research group faced several challenges stemming from both the ocean environment itself and the long distance between the ocean platform and the shore. The topside system antenna may lose line of sight (LOS) with the shore system antenna due to the Earth curvature, platform tilt, and wave height (Figure 1). Prior to finalizing the OWL manager design, we also performed a risk assessment of the OCTT's Prognosis and Health Monitoring (PHM) system using the SHIELD design analysis method ([6, 7]) to create a more robust and resilient all around system architecture.

The results from our SHIELD evaluation indicated that adding a second, redundant wireless link lowered the probability of communication failure by approximately 4.6%. The packet flows on the two redundant radio links (cellular and FreeWave [8]) require autonomous management for capacity allocation and packet scheduling in order to prevent QoS degradation when the link capacity is variable.

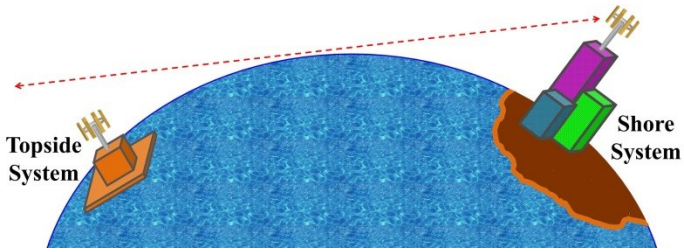

Figure 1: ***Simplified example of a Topside system antenna losing line of sight with the shore due to wave height and platform angle.***

The topside system [9], shown in Figure 2, uses several custom-designed systems to autonomously monitor OCTT system health and boost its overall robustness and resilience. Condition Based Maintenance (CBM) is utilized to monitor the mechanical components (i.e. turbine, generator, gearbox, shafts, housings) of the ocean platform with the objective of reducing the probability of system failure [10] while extending the interval between preventative maintenance services for prototype commercial variants. Using the real time sensor data, CBM forecasts the health of the system and can send control commands to modify system parameters or shut down the system to prevent expensive damage to hardware when anomalous conditions are imminent.

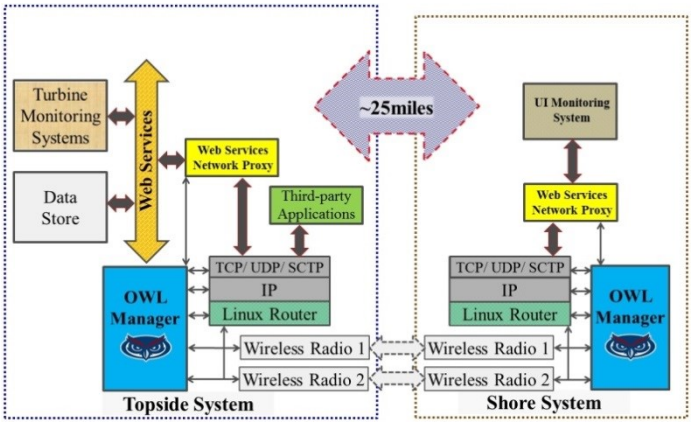

Figure 2: ***Architecture of the OCTT communication system and general CBM components.***

The bandwidth limitations of the wireless links prevent the transmission of all data generated by OCTT sensors and monitoring systems. Transmitting megabit/s data flows would cause high delay, congestion, and packet loss. Therefore, the ocean-side component of the PHM system processes the data locally and then stores the resulting values in the Data Store (DS) component. The DS consists of a MySQL database and MIMOSA OSA-CBM [11, 12] compliant SOAP/XML Web Services (WS) [9] implemented on Apache Tomcat with Java EE. External interaction with the system from the shore may also be initiated through the DS interface.

The DS responds to client requests using SOAP/XML messages. These replies pass through the Web Services Network Proxy component, part of our wireless QoS architecture. This assigns each reply a priority based on message semantics at the OSA-CBM layer, such as client identity, data/sensor type, timeliness. The proxy compresses SOAP/XML [13] replies to further reduce the load over the wireless link. Compression ratios above 50% are typical for these messages. This topside proxy is implemented as a Java servlet operating at the application layer and cannot use any low level TCP/IP and MAC QoS techniques described in [14, 15]. Since at the IP layer all HTTP/TCP packets look the same, the Queue Manager must control TCP segment scheduling for SOAP replies based on their semantics. This is necessary to maintain application-level QoS over unreliable wireless links. Without this application-aware QM scheduling, SOAP reply packets would fill up TCP buffers, with FIFO scheduling that causes unpredictable delays and priority inversion when congestion occurs due to variable link quality.

The QM schedules SOAP reply message transmission by controlling when message blocks are sent back to the clients on different servlet threads, using a one-thread-per-request model. After transmitting each data block, each thread is forced to sleep, maintaining an acceptable average data rate for the wireless link according to application QoS policies and the link quality forecast received from the OWL manager. The OWL manager does its data rate forecast by applying sensor fusion and machine learning using various sensor data streams onboard the OCTT. It accumulates periodically local and environmental measurements for a specific time interval. After that, OWL generates a new forecasting model. In our initial tests, we focused on predicting the channel quality (received power and data rate). We consider that the main channel fading factor is antenna line of sight (LOS) loss due to the Earth's curvature and wave movement which changes the topside antenna height. We report on forecasting performance based on data rate and link margin models determined for the FreeWave HTP [8] radios.

The rest of this paper is organized as follows. In section II, we brief discuss some related work. Section III describes the OWL management framework. The experimental results and prototype evaluation are discussed in section IV and section V concludes the paper with a discussion of the results and future work.

II. Related Work

Prior to designing our link management framework for the OCTT, we investigated several related projects to gain a better understanding of possible problems and challenges we might face. Furthermore, we also were able to observe and learn what solutions worked best for their specific application environment. This section details several methods of wireless link management, link state prediction, scheduling, and flow control.

In [16] the authors present SoftRate, a wireless bit rate adaptation protocol that is responsive to rapidly varying channel conditions. SoftRate uses confidence information calculated by the physical layer which is exported to higher layers via the SoftPHY interface to estimate the prevailing channel bit error rate (BER) and differs from earlier work that used frame receptions or SNR estimates to select bit rates. The transmitter uses this BER estimate which is calculated for each received packet, regardless of error, to choose the optimal choice of bit rate. SoftRate's novel BER computation works in a variety of wireless environments, using diverse

hardware, without having to retrain itself. Their solution uses abrupt changes in the BER estimate to identify interference which enables it to reduce the bit rate only in response to channel errors caused by attenuation or fading. They conducted experiments using a simulated radio prototype to show that their solution achieves double the throughput of other frame level protocols such as SampleRate and RRAA. It also achieved ~20% more throughput than an SNR-based protocol trained on the same operating environment and ranged up to four times greater throughput than an untrained SNR-based protocol.

In [17] the authors have conducted a systematic measurement based study to confirm that in general SNR is a good prediction tool for channel quality and identified two key challenges. The SNR measures for each type of hardware are often not calibrated even though SNR thresholds *are* hardware dependent. Secondly, predictions for the correlations of SNR to frame delivery ratio (FDR) typically is over optimistic under interference conditions. Considering these two challenges the authors present a practical SNR Guided Rate Adaptation (SGRA) scheme. Furthermore, they stress that rate adaptation is critical to the system performance of wireless networks. Typically rate adaptation is considered as a MAC layer mechanism in IEEE 802.11. Similar projects rely only on frame losses to infer channel quality and will perform poorly if frame losses are caused by interference. Other SNR based rate adaptation schemes have been proposed but most of them have not been implemented in the real world. The authors implement and evaluate SGRA in a real testbed and compare its' performance with ARF, RRAA and HRC. In all their test cases the results show that SGRA outperformed the other three algorithms.

The authors in [18] develop a cross-layer design for multi-user scheduling at the data link layer with each user employing adaptive modulation and coding (AMC) at the physical layer. By classifying users into either QoS guaranteed users and best effort users their proposed scheduler enables both prescribed QoS guarantees and efficient bandwidth utilization simultaneously. Furthermore, the cross-layer scheduler uses a simplistic implementation, provides service isolation and scalability, decouples delay from dynamically scheduled bandwidth and is backward compatible with existing separate layer designs. The performance results in the analysis are verified by various simulations and pertinent issues with regard to robustness. They use numerical examples to illustrate the steady state statistical performance for both single and multiple users and show the asymptotic behavior of their design for a large number of users. First they derive the QoS of a certain user in terms of throughput, packet loss rate and average delay for a given bandwidth allocation. They then proposed an algorithm for determining the minimal required bandwidth to ensure the prescribed QoS.

III. OWL MANAGEMENT FRAMEWORK

The primary objectives of the OWL manager are 1) support queue management and transmission scheduling of XML sensor data over HTTP by the Topside Proxy with an accurate wireless link capacity forecast, and 2) manage IP packet queuing and capacity allocation on the main router connecting the topside network with the wireless links to the shore. OWL taps into the OCTT's real time sensor data streams (GPS, analog, IMU, etc.) and in combination with the wireless link's received signal strength indicator (RSSI) reading it builds periodically a machine learning model that is used to forecast the RSSI for a limited time window. The RSSI for the shore-ocean link is a primary predictor for the link quality on the channel in the reverse direction being used for ocean-to-shore data delivery. Using the RSSI on the topside node to predict data rate in the opposite direction avoids the need to implement a feedback protocol involving RSSI monitoring at the shore receiver and forwarding the link quality to the topside OWL system. Hence, reaction time to abrupt changes in link quality is reduced. Based on the sensed platform attributes and offline measurements that established the data rate dependency on the link margin, OWL calculates the signal power and throughput of the link.

OWL integrates seamlessly with the QM to fine tune the QoS and ensure that the transmission data rate never exceeds the capacity of the wireless radios. Without this knowledge, the amount of data released by the QM may unknowingly exceed the amount of data that the wireless link may transmit, resulting in buffer overflows and eventual timeout of the messages from the Tomcat server when the TCP timeout limit is reached. Dropped packets from the Topside Proxy may consequently cause a cascade of other errors possibly as distant as the Shore Proxy which may result in timeouts for clients or complete retransmissions of data. OWL assists in the prevention of these errors by forecasting the throughput of the link. The forecast is used to throttle the flows of data to the two topside wireless radios preventing buffer overflows. Additionally, each periodic forecast is sent to the QM which may then prioritize accordingly the data sent to the router for transmission.

OWL manages the flow of data over the redundant link, shown in Figure 2, from the IP layer using the *iproute2* [19, 20] package. *iproute2* is a suite of Linux based, command line utilities which manipulate kernel structures for IP layer network configuration. Of the tools in the *iproute2* package, *tc* [21] is used for traffic control and it allows OWL to perform all of the configuration of the kernel structures necessary to support and manage traffic over the two radios. The *iproute* tool from the *iproute2* package is used to manage and manipulate the routing tables of the network.

The normal operation of OWL is such that it continuously loops through two primary tasks, capture, where sensory data is gathered and recorded, and modeling and prediction, where time series analysis is used to forecast a periodic interval of the wireless link throughput based on platform attribute data. Together these tasks help OWL maintain an accurate forecast of the wireless link and shape the traffic over the wireless radios. The first task, capture, is continuous and provides real time instance data for the machine learning time series analysis. OWL applies data fusion to gather the forecasting sensor metrics of the platform from the sensory data streams of the OCTT. This incoming data is continuously captured from

several sensing components which include the Three-axis Compass Module (TCM), Inertial Measurement Unit (IMU), and GPS. Current values for RSSI are obtained from the wireless transceiver.

The candidate technologies for the OCTT wireless links are the FreeWave HTP radio [8] and 4G cellular USB modems. Experiments from 2010 with a 3G cellular modem have shown that data connectivity is available at ranges exceeding 25 km off the coast of Fort Lauderdale, Florida.

To test the OWL manager we generated a synthetic RSSI trace from which we derived the channel data rate. We assumed a worst case scenario when the platform is moored near the edge of the line-of-sight from the shore antenna, implying that variation in the topside antenna height may intermittently obscure LOS. This causes fading that is magnified by strong (high) ocean waves. This situation would benefit the most from QoS adaptation and link capacity allocation mechanisms, in contrast with the case when the wireless link has constant LOS and thus a solid signal quality the majority of the time. Reducing the distance between transceivers and increasing antenna heights contribute to reducing the LOS loss probability in both directions.

Platform data streams with platform information are combined with the current link quality data (RSSI) and history are passed to the link forecasting component. The FreeWave HTP radio has an SNMP interface that provides statistics on current and average signal-to-noise ratio (SNR), bit error rate (BER), data rate, and other link metrics. Cellular USB modems offer the AT command interface that delivers similar information.

For development and early prototype evaluation before sea trials with the wireless links are performed we relied on earlier recorded platform sensor data streams for position (latitude/longitude), heading, altitude, and 3D accelerometer. We generated synthetic traces for the received signal power (RSSI) at periodic times (100 Hz) using a knife edge diffraction model to approximate the effects of antenna height variation near the LOS limit [32]. The power budget considered is:

$$P_{rxdB} = P_{txdB} + G_{FSdB} + G_{a1dB} + G_{a2dB} + G_{ddB} + G_{pdB}$$

where the parameters are described in the following table:

P_{txdB}	radio transmit power	30 dBm
G_{FSdB}	free space path gain	-141dB
G_{a1dB}	shore antenna gain	12.15 dB
G_{a2dB}	ocean platform antenna gain	8 dB
G_{ddB}	diffraction gain that varies depending on antenna height	dependent on antenna height
G_{pdB}	antenna polarization mismatch gain	dependent on antenna angles

The link margin (difference between P_{rxdB} and the receiver sensitivity (-102 dBm)) then determines the link data rate using interpolation from a table provided by the manufacturer.

We plan to conduct further experiments that capture the RSSI and throughput directly from the wireless radios in real time while at sea.

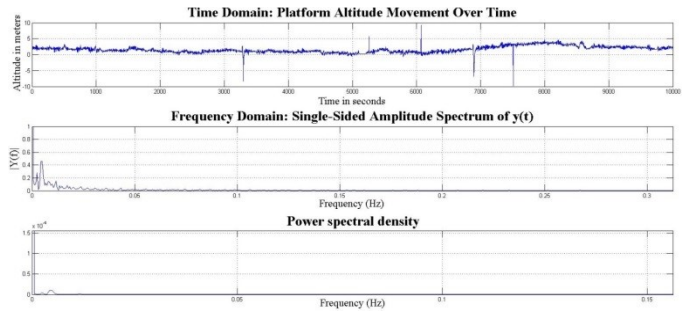

Figure 3: ***Time Domain, FFT, and power spectral density of the platform movement after smoothing the incoming data.***

To determine the window size for past sensor data history needed for forecast we conducted a Fourier analysis of the platform altitude data. We considered that ocean wave movement is the primary factor affecting the link quality due to antenna height variation. From the FFT power spectral density plot (Figure 3) the waves with higher amplitudes have periods of about 30 seconds. The link forecast thus uses the most recent 30 sensor data history to build the forecast model.

The high frequency platform sensor data sampled at 100 Hz had certain noise. We smoothed the data using a sliding window filter, computing the mean value for 100 samples every second.

The resulting output had a reduction in the noise levels which may be seen in Figure 3. From this smoothing, the periodic nature of the waves becomes clearly visible, displaying the high frequency surface waves and the low frequency swell.

After initially capturing five minutes of platform movement and channel conditions, OWL executes the second task, modeling & prediction. This task generates a time series forecasting model based on the incoming sensory data using the Weka time series forecasting library [27].

The generic series forecasting algorithm removes the time dependency for target attributes by creating new attributes which hold the values of the target at previous time steps. These are called “lagged” variables. In our case, we predict the data rate and this attribute is expanded by Weka's time series forecasting algorithm into additional attributes with values from past time steps, datarate@time-1, datarate@time-2, datarate@time-3,...

The time series forecasting algorithm then may use a classifier (for symbolic classes) or a regression algorithm (for numeric data) on the expanded data set that includes the generated lagged variables.

The forecast window size indicates how far into the future predicted data is computed. Prediction error grows with a longer window, as expected. The OWL framework sends each data rate forecast window to the Queue Manager that adjusts transmission schedule for web service threads. If a drop in data rate is predicted, then the QM delays transmissions to prevent congestion. Conversely, if the data rate increases, the QM will

allow more traffic from the application layer by waking threads according to priority as described in [25]. OWL continually updates the model to provide an up to date and accurate representation of the surface conditions of the oceanic environment and their current effect on the OCTT communication system.

IV. PROTOTYPE EVALUATION

Our initial experimentation of the system was conducted using several Linux based computers emulating the nodes of the OCTT network. To accurately emulate the varying conditions of the wireless link we used NetEM [26]. NetEM consists of two main components, a tiny kernel module for a queuing discipline and a utility for its configuration which permits the emulation of such link effects as packet delay, loss, duplication, and corruption.

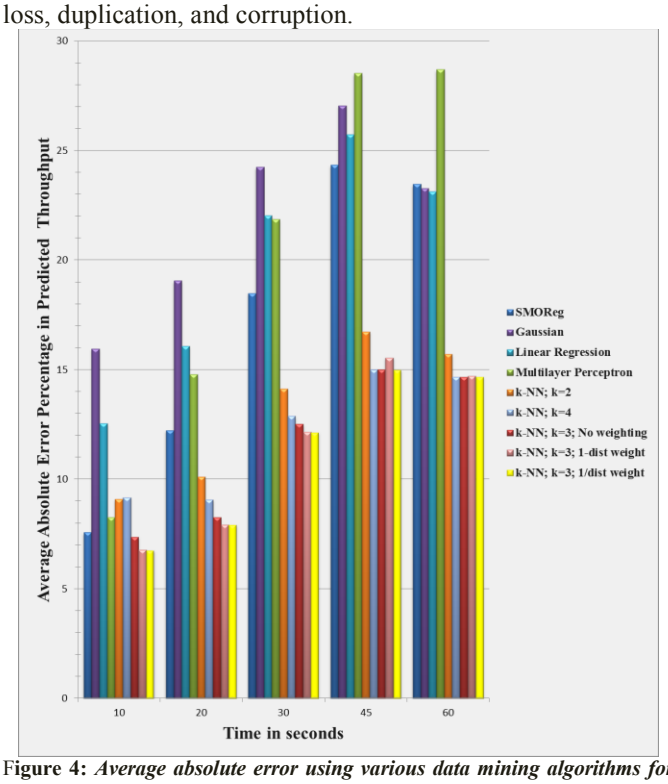

Figure 4: ***Average absolute error using various data mining algorithms for the first 60 second throughput forecast. As k-NN was the best overall performer, we also show the results for varying the k value and weighting methods.***

For our preliminary testing of the prototype, using a 1.1 million instance dataset gathered from the five incoming sensory data streams of the OCTT, we used the Weka toolkit [27] to simplify preliminary tests using five different time series forecasting algorithms including k-NN [24], SMOReg [28], Gaussian Processes [29], Linear Regression [30], and Multilayer Perceptron [31]. In Figure 4, we show the average absolute error values using these algorithms at various forecasted time units in seconds. These early experiments showed that the best performer, k-NN, forecasts on the average a value within ~8kbps of the actual throughput in the first 10 seconds. We used the platform altitude because loss of line of sight between the two antennas becomes more of an issue in the oceanic environment as is easily seen in Figure 1 and the distance between the main components which places the ocean platform at the boundary limits of the wireless communication system.

After determining the best k value, we tested the various methods of weighting the distance; no weighting, 1–distance, and 1/distance. Experimental results show that weighting the distances increased the precision as opposed to no weighting. In Figure 5 the result for each predicted value using 1–distance weighting method provides visual confirmation that there was very little difference between the actual and predicted data rate over the 1.1 million instance dataset. Furthermore, our preliminary testing demonstrated that k-NN, with $k = 3$ using 1-distance weighting predicts the link throughput with 84.09% accuracy with an average absolute error rate of 15.909%.

Again in Figure 5, it is important to note the relatively smooth, periodic wave form of the predicted values using distance weighting with k-NN which had fewer random spikes in the forecast and also that the trends in the predicted values logically reflect the periodic throughput fluctuations experienced by a wireless link based on an anchored, free body object in an oceanic environment. Initial tests with k-NN were conducted with $k = 1$, and required further experimentation changing the k value and distance weighting method, to gain the most precise forecasting model. The parameters of k-NN (i.e. number nearest neighbors, weighting, and distance) may require further testing and refinement prior to the final deployment of the OCTT into the environment.

V. CONCLUSIONS

In this article we have described the major components of the topside proxy and PHM architectures including the mechanisms used by the QM. The OWL manager provides an autonomous means to control the queue management for the topside system. Our objective in its development was to override the TCP layer control artificially from the application layer. This allows full control over the buffers which may otherwise suffer from congestion or overflow under heavy loads while also avoiding the operational limitations associated with TCP such as flow and retransmission control.

OWL's forecasting capability allows us to predict the throughput of the wireless link based on historical PHM sensor data. Our results from preliminary testing with various DMML algorithms revealed the best choice to be k-NN which proved to be highly accurate in predicting the data rate in the first ten seconds and retaining nearly the same accuracy in a 30 second window. Upon predicting that the link throughput will drop below a specified level, OWL issues an alert to the Queue Manager to stop all transmission to the TCP layer. Once an adequate data rate is forecasted again, a control is sent to continue transmission.

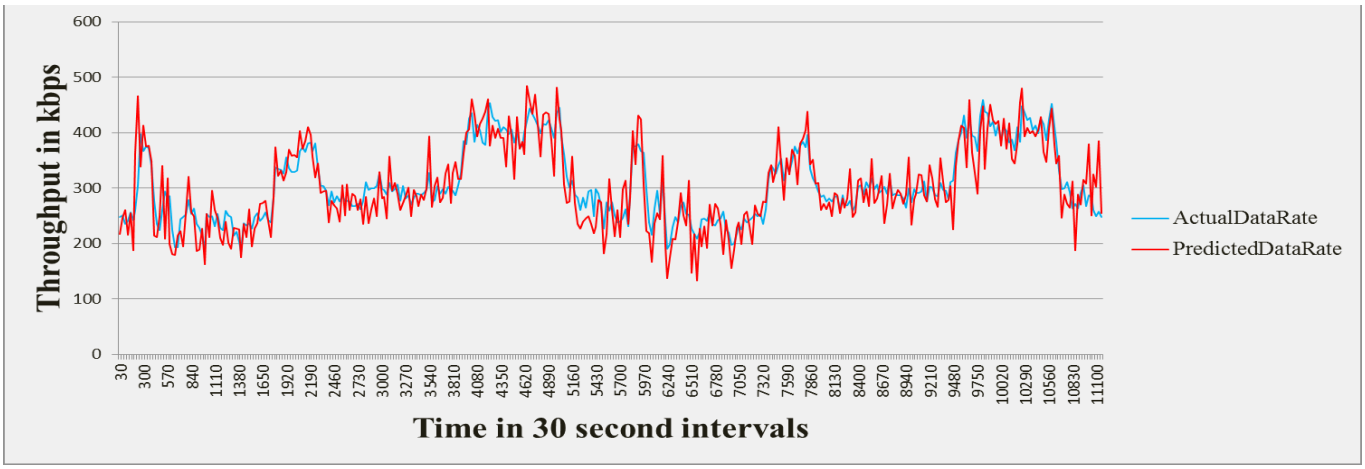

Figure 5: ***Comparison of the actual and forecasted throughput values using the k-NN algorithm, k = 3, and distance weighting over the full dataset forecast. This chart clearly shows the accuracy of the predictions calculated using 1-distance weighting for k-NN and the periodic 'up and down' movement of waves affecting the throughput of a platform based wireless link in the oceanic environment.***

The OWL manager provides an effective means to assist the QM of the topside system. Our objective in its development was to manually override the TCP layer control, artificially from the application and IP layers. This allows full control over the buffers which may otherwise suffer from congestion or overflow under heavy loads while avoiding operational limitations associated with TCP such as flow rate and retransmission control. Furthermore, OWL manages the flow of data over the redundant wireless radios such that optimal throughput and load balancing is consistently maintained.

REFERENCES

[1] "Tidal Power." Generating Electricity from Tidal Currents. N.p., n.d. Web. 10 Sept. 2013.

[2] "Ocean Power." Energy Report. N.p., n.d. Web. 10 Sept. 2013.

[3] "Chapter 1. Renewable Energy." Renewable Energy Overview. N.p., n.d. Web. 10 Sept. 2013.

[4] "The Florida Current." The Florida Current. N.p., n.d. Web. 11 Sept. 2013.

[5] Cardei, I., A. Agarwal, B. Alhalabi, T.Tavtilov, T. Khoshgoftaar, and P. Beaujean, "Software and Communications Architecture for Prognosis and Health Monitoring of Ocean-based Power Generator." (2011): 1-8.

[6] Marcus, A., I. Cardei, and T. Tavtilov. "Resilient System Design for Prognosis and Health Monitoring of an Ocean Power Generator." IEEE Systems Conference 2012 (2012): 1-8. IEEE, 2012.

[7] Marcus, A., I. Cardei, and G. Alsenas. "Automation of the SHIELD Methodology for System Hazard Analysis and Resilient Design." IEEE Systems Conference 2013 (2013): 1-8. IEEE, 2013.

[8] FreeWave Technologies, Boulder, Colorado, <http://www.freewave.com>

[9] Tavtilov, T. "Prognosis and Health Monitoring Communications Quality Of Service." Thesis (M.S.C.S.)--Florida Atlantic University (2011): 1-98. 2011.

[10] Kostyukov, V. "Safe Maintenance of the Equipment is the Technology of the 21st Century - Russian Experience", SPC Dynamics, Omsk, Russia, 1999.

[11] Lee, Kang, and Robert X. Gao. "Sensor Network and Information Interoperability - Integrating IEEE 1451 with MIMOSA and OSA-CBM." IEEE, 2002.

[12] Lebold, M., K. Reichard, and S. Byington. "OSA-CBM Architecture Development with Emphasis on XML Implementations." MARCON, 2002.

[13] "Java Technology & XML Downloads - SOAP with Attachments API for Java (SAAJ)."Oracle Technology Network for Java Developers.

[14] Firoiu, V., Le Boudec, J., Towsley, D., Zhang, Z., "Theories and models for Internet quality of service," Proceedings of the IEEE, vol.90, no.9, pp. 1565-1591, Sep 2002.

[15] Fragouli, ChristineC., Vijay, SivaramanS., and Mani B. Srivastava, M., "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling.", InfoCOM, 1998.

[16] Vutukuru, M., H. Balakrishnan, and K. Jamieson. "Cross-Layer Wireless Bit Rate Adaptation." SigComm (2009): 1-13. ACM, Aug. 2009.

[17] Zhang, J., K. Tan, J. Zhao, H. Wu, and Y. Zhang. "A Practical SNR-Guided Rate Adaptation." INFOCOM (2008): 1-5. Microsoft Research Asia and the IEEE Communications Society, 2008.

[18] Liu, Q., S. Zhou, and G.B. Giannakis. "Cross-Layer Scheduling With Prescribed QoS Guarantees in Adaptive Wireless Networks." IEEE Journal on Selected Areas in Communications 23.5 (2005): 1056-066. IEEE, May 2005.

[19] "IPRoute2." The Linux Foundation. N.p., 19 Nov. 2009. Web. <http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2>.

[20] Hubert, B. "Linux Advanced Routing & Traffic Control HOWTO." Linux Advanced Routing & Traffic Control HOWTO. N.p., n.d. Web. <http://lartc.org/howto/index.html>.

[21] Hubert, B. "TC." TC. Linuxmanpages.com, n.d. Web. 07 Sept. 2011. <http://www.linuxmanpages.com/man8/tc.8.php>.

[22] Hubert, B. "Linux Advanced Routing & Traffic Control HOWTO." Linux Advanced Routing & Traffic Control HOWTO. N.p., n.d. Web. <http://lartc.org/howto/index.html>.

[23] Hubert, B. "TC." TC. Linuxmanpages.com, n.d. Web. 07 Sept. 2011. <http://www.linuxmanpages.com/man8/tc.8.php>.

[24] "IBk." IBk. Weka, n.d. Web. 16 Sept. 2013.

[25] Tavtilov, T. "Prognosis and Health Monitoring Communications Quality Of Service." Thesis (M.S.C.S.)--Florida Atlantic University (2011): 1-98. 2011.

[26] "NetEM." The Linux Foundation. N.p., 19 Nov. 2009. Web. <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>.

[27] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." Computer Science Department, University of Waikato. Web. 5 Sept. 2011. <http://www.cs.waikato.ac.nz/ml/weka/>.

[28] "SMOreg." SMOreg. Weka, n.d. Web. 16 Sept. 2013.

[29] "GaussianProcesses." GaussianProcesses. Weka, n.d. Web. 16 Sept. 2013.

[30] "Introduction." Weka. WekaWiki, n.d. Web. 16 Sept. 2013.

[31] "Multilayer Perceptron." MultilayerPerceptron. Weka, n.d. Web. 16 Sept. 2013.

[32] T. Rappaport, "Wireless Communications: Principles and Practice (2nd Edition)", Prentice Hall; 2 edition