

Fall 10-18-2012

ENSURE: A Time Sensitive Transport Protocol to Achieve Reliability Over Wireless in Petrochemical Plants

Leigh Sheneman

Follow this and additional works at: https://scholarworks.uttyler.edu/compsci_grad

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Sheneman, Leigh, "ENSURE: A Time Sensitive Transport Protocol to Achieve Reliability Over Wireless in Petrochemical Plants" (2012). *Computer Science Theses*. Paper 4.
<http://hdl.handle.net/10950/94>

This Thesis is brought to you for free and open access by the School of Technology (Computer Science & Technology) at Scholar Works at UT Tyler. It has been accepted for inclusion in Computer Science Theses by an authorized administrator of Scholar Works at UT Tyler. For more information, please contact tbianchi@uttyler.edu.

ENSURE: A TIME SENSITIVE TRANSPORT PROTOCOL TO
ACHIEVE GUARANTEED RELIABILITY OVER WIRELESS
PETROCHEMICAL PLANTS

by

LEIGH SHENEMAN

A thesis submitted in partial fulfillment of the requirements for
the degree of
Master of Science
Department of Computer Science

Arun Kulkarni, PhD., Committee Chair

College of Engineering and Computer Science

The University of Texas at Tyler
September 2012

The University of Texas at Tyler
Tyler, Texas

This is to certify that the Master's Thesis of

LEIGH SHENEMAN

has been approved for the thesis requirement on

September 6, 2012
for the Master of Science degree

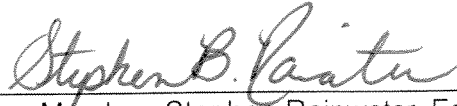
Approvals:



Thesis Chair: Arun Kulkarni, Ph.D.



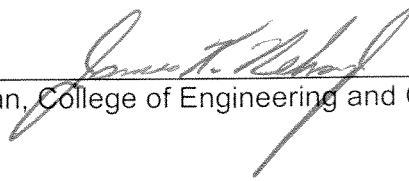
Member: Narayanan Subramanian, Ph.D.



Member: Stephen Rainwater, Ed.D.



Chair, Department of Computer Science



Dean, College of Engineering and Computer Science

© Copyright by Leigh Sheneman 2012
All rights reserved

Table of Contents

List of Tables.....	iv
List of Figures	v
Abstract.....	v
Chapter I - Introduction	1
Chapter II - Related Work	5
Chapter III - Timely and Guaranteed Packet Transmission	10
3.1 Introduction.....	10
3.2 Defining Basic Equations	12
3.3 Addressing Reliability	14
3.4 Meeting a Deadline	19
3.5 The Best of Both Worlds	23
Chapter IV - Approach	28
4.1 Refining Typology.....	28
4.2 Determining Network Connections	30
4.3 Protocol Placement	31
4.4 Simulation Models	32
4.4.1 ENSURE 1.0.....	32
4.4.2 ENSURE 2.0.....	33

4.4.3 ENSURE 3.0	33
Chapter V - Experimental Results.....	34
5.1 Introduction.....	34
5.2 ENSURE 1.0	34
5.2.1 Delay versus Channel Utilization	35
5.2.2 Noise versus Channel Utilization	39
5.2.3 Channel Utilization versus Reliability	43
5.2.4 Noise versus Retransmissions	44
5.2.5 Reliability versus Retransmissions	46
5.2.6 Noise versus Packet Loss	48
5.2.7 Noise versus Domain of Delay and Channel Utilization	48
5.3 ENSURE 2.0: Channel Utilization	52
5.3.1 Delay versus Channel Utilization	52
5.3.2 Noise versus Channel Utilization	56
5.3.3 Channel Utilization versus Reliability	60
5.3.4 Noise versus Retransmissions	64
5.3.5 Reliability versus Retransmissions	68
5.3.6 Noise versus Packets Loss.....	71
5.3.7 Reliability versus Packet Loss	74

5.3.8 Noise versus Domain of Delay and Channel Utilization	77
5.3.9 Reliability versus Domain of Delay and Channel Utilization	81
5.4 ENSURE 3.0: Error Correction	85
5.4.1 Delay versus Channel Utilization	85
5.4.2 Noise versus Channel Utilization	89
5.4.3 Channel Utilization versus Reliability	92
5.4.4 Noise versus Retransmissions	93
5.4.5 Reliability versus Retransmissions	97
5.4.6 Noise versus Packets Loss	99
5.4.7 Noise versus Domain of Delay and Channel Utilization	99
5.4.8 Reliability versus Domain of Delay and Channel Utilization	103
5.5 Overall Transport Protocol Improvements	107
5.5.1 Delay versus Channel Utilization	107
5.5.2 Noise versus Channel Utilization	111
5.5.3 Noise versus Packets Loss	115
5.5.4 Reliability versus Packet Loss	119
Chapter VI - Conclusions and Future Research	123
Works Cited	125

List of Tables

Table 1. Channel utilization versus reliability for one sensor using ENSURE 1.0	43
Table 2. Channel utilization versus reliability for two sensors using ENSURE 1.0	44
Table 3. Channel utilization versus reliability for three sensors using ENSURE 1.0	44
Table 4. Reliability versus retransmissions for one sensor using ENSURE 1.0	47
Table 5. Reliability versus retransmissions for two sensors using ENSURE 1.0	47
Table 6. Reliability versus retransmissions for three sensors using ENSURE 1.0	48
Table 7. Channel utilization versus reliability for one sensor using ENSURE 3.0.	92
Table 8. Channel utilization versus reliability for two sensors using ENSURE 3.0	93
Table 9. Channel utilization versus reliability for three sensors using ENSURE 3.0	93
Table 10. Reliability versus retransmissions for one sensor using ENSURE 3.0	97
Table 11. Reliability versus retransmissions for two sensors using ENSURE 3.0	98
Table 12. Reliability versus retransmissions for three sensors using ENSURE 3.0	98

List of Figures

Figure 1. Operation cost coefficient compared to number of required retransmissions..	15
Figure 2. Finite state machine for ENSURE 1.0 – intermediate node.....	17
Figure 3. Finite state machine for ENSURE 1.0 – base station	18
Figure 4. Algorithm for ENSURE 1.0	19
Figure 5. Finite state machine for ENSURE 2.0 – intermediate node.....	21
Figure 6. Finite state machine for ENSURE 2.0 – base station	22
Figure 7. Algorithm for ENSURE 2.0	23
Figure 8. Finite state machine for ENSURE 3.0 – intermediate node.....	25
Figure 9. Finite state machine for ENSURE 3.0 - base station	26
Figure 10. Algorithm for ENSURE 3.0	27
Figure 11. Sample of network topology currently used	29
Figure 12. Sensor typology with ENSURE implemented	30
Figure 13. Delay versus channel utilization for one sensor using ENSURE 1.0.....	36
Figure 14. Delay versus channel utilization for two sensors using ENSURE 1.0	36
Figure 15. Delay versus channel utilization for three sensors using ENSURE 1.0.....	37
Figure 16. Comparing delay versus channel utilization for all sensors using ENSURE 1.0.....	38
Figure 17. Noise versus channel utilization for one sensor using ENSURE 1.0.....	40
Figure 18. Noise versus channel utilization for two sensors using ENSURE 1.0	40
Figure 19. Noise versus channel utilization for three sensors using ENSURE 1.0.....	41
Figure 20. Noise versus channel utilization for all simulations using ENSURE 1.0	42

Figure 21. Noise versus retransmissions for one sensor when using ENSURE 1.0.....	45
Figure 22. Noise versus retransmissions for two sensors when using ENSURE 1.0	46
Figure 23. Noise versus retransmissions for three sensors when using ENSURE 1.0...	46
Figure 24. Noise versus domain of delay and channel utilization with one sensor using ENSURE 1.0	49
Figure 25. Noise versus domain of delay and channel utilization with two sensors using ENSURE 1.0	50
Figure 26. Noise versus domain of delay and channel utilization with three sensors using ENSURE 1.0	51
Figure 27. Delay versus channel utilization for one sensor using ENSURE 2.0	53
Figure 28. Delay versus channel utilization for two sensors using ENSURE 2.0	54
Figure 29. Delay versus channel utilization for three sensors using ENSURE 2.0	54
Figure 30. Comparison of delay versus channel utilization for all sensors using ENSURE 2.0	55
Figure 31. Noise versus channel utilization for one sensor using ENSURE 2.0	57
Figure 32. Noise versus channel utilization for two sensors using ENSURE 2.0	57
Figure 33. Noise versus channel utilization for three sensors using ENSURE 2.0	58
Figure 34. Comparing noise versus channel utilization for all sensors using ENSURE 2.0	59
Figure 35. Reliability versus channel utilization for one sensor using ENSURE 2.0.....	61
Figure 36. Reliability versus channel utilization for two sensors using ENSURE 2.0	61
Figure 37. Reliability versus channel utilization for three sensors using ENSURE 2.0...	62
Figure 38. Comparison of reliability versus channel utilization for all sensors using ENSURE 2.0	63
Figure 39. Noise versus number of retransmissions required for one sensor using ENSURE 2.0	65

Figure 40. Noise versus number of retransmissions required for two sensors using ENSURE 2.0	65
Figure 41. Noise versus number of retransmissions required for three sensors using ENSURE 2.0	66
Figure 42. Comparison of noise versus number of retransmissions required for all sensors using ENSURE 2.0.....	67
Figure 43. Reliability versus number of retransmissions required for one sensor using ENSURE 2.0	68
Figure 44. Reliability versus number of retransmissions required for two sensors using ENSURE 2.0	69
Figure 45. Noise versus number of retransmissions required for three sensors using ENSURE 2.0	69
Figure 46. Comparison of noise versus number of retransmissions required for all sensors using ENSURE 2.0.....	70
Figure 47. Noise versus number of packets lost for two sensors using ENSURE 2.0....	71
Figure 48. Noise versus number of packets lost for three sensors using ENSURE 2.0 .	72
Figure 49. Comparison of noise versus number of packets lost for all sensors using ENSURE 2.0	73
Figure 50. Reliability versus number of packets lost for two sensors using ENSURE 2.0	75
Figure 51. Reliability versus number of packets lost for three sensors using ENSURE 2.0	75
Figure 52. Comparison of reliability versus number of packets lost for all sensors using ENSURE 2.0	76
Figure 53. Noise versus domain of delay and channel utilization for one sensor using ENSURE 2.0	78

Figure 54. Noise versus domain of delay and channel utilization for two sensors using ENSURE 2.0	79
Figure 55. Noise versus domain of delay and channel utilization for three sensors using ENSURE 2.0	80
Figure 56. Reliability versus domain of delay and channel utilization for one sensor using ENSURE 2.0	82
Figure 57. Reliability versus domain of delay and channel utilization for two sensors using ENSURE 2.0	83
Figure 58. Reliability versus domain of delay and channel utilization for three sensors using ENSURE 2.0	84
Figure 59. Delay versus channel utilization for one sensor using ENSURE 3.0	86
Figure 60. Delay versus channel utilization for two sensors using ENSURE 3.0	86
Figure 61. Delay versus channel utilization for three sensors using ENSURE 3.0	87
Figure 62. Comparison delay versus channel utilization for all sensors using ENSURE 3.0	88
Figure 63. Noise versus channel utilization for one sensor using ENSURE 3.0	89
Figure 64. Noise versus channel utilization for two sensors using ENSURE 3.0	90
Figure 65. Noise versus channel utilization for three sensors using ENSURE 3.0	90
Figure 66. Comparison noise versus channel utilization for all sensors using ENSURE 3.0	91
Figure 67. Noise versus retransmissions for one sensor using ENSURE 3.0.	94
Figure 68.Noise versus retransmissions for two sensors using ENSURE 3.0.....	95
Figure 69. Noise versus retransmissions for three sensors using ENSURE 3.0.	95
Figure 70. Comparison of noise versus retransmissions for all sensors using ENSURE 3.0.....	96

Figure 71. Noise versus domain of delay and channel utilization for one sensor using ENSURE 3.0	100
Figure 72. Noise versus domain of delay and channel utilization for two sensors using ENSURE 3.0	101
Figure 73. Noise versus domain of delay and channel utilization for three sensors using ENSURE 3.0	102
Figure 74. Reliability versus domain of delay and channel utilization for one sensor using ENSURE 3.0	104
Figure 75. Reliability versus domain of delay and channel utilization for two sensors using ENSURE 3.0	105
Figure 76. Reliability versus domain of delay and channel utilization for three sensors using ENSURE 3.0	106
Figure 77. Protocol comparison of delay versus channel utilization for one sensor	108
Figure 78. Protocol comparison of delay versus channel utilization for two sensors....	109
Figure 79. Protocol comparison of delay versus channel utilization for three sensors .	110
Figure 80. Protocol comparison for noise versus channel utilization of one sensor	112
Figure 81. Protocol comparison for noise versus channel utilization of two sensors....	113
Figure 82. Protocol for noise versus channel utilization of three sensors	114
Figure 83. Protocol comparison of noise versus packet loss for one sensor	116
Figure 84. Protocol comparison of noise versus packet loss for two sensors	117
Figure 85. Protocol comparison of noise versus packet loss for three sensors	118
Figure 86. Protocol comparison of reliability versus packet loss for one sensor	120
Figure 87. Protocol comparison of reliability versus packet loss for two sensors	121
Figure 88. Protocol comparison of reliability versus packet loss for three sensors	122

Abstract

ENSURE: A TIME SENSITIVE TRANSPORT PROTOCOL TO ACHIEVE GUARANTEED RELIABILITY OVER WIRELESS PETROCHEMICAL PLANTS

Leigh Sheneman

Thesis Chair: Arun Kulkarni, Ph.D.

The University of Texas at Tyler
December 2012

As society becomes more reliant on the resources extracted in petroleum refinement the production demand for petrochemical plants increases. A key element is producing efficiently while maintaining safety through constant monitoring of equipment feedback. Currently, temperature and flow sensors are deployed at various points of production and 10/100 Ethernet cable is installed to connect them to a master control unit. This comes at a great monetary cost, not only at the time of implementation but also when repairs are required. The capability to provide plant wide wireless networks

would both decrease investment cost and downtime needed for repairs. However, the current state of wireless networks does not provide any guarantee of reliability, which is critical to the industry. When factoring in the need for real-time information, network reliability further decreases.

This work presents the design and development of a series of transport layer protocols (coined ENSURE) to provide time-sensitive reliability. ENSURE 1.0 has a pure focus on reliability and time was not considered. The first objective was to meet 100% reliability in information delivery by using proactive redundant data transmissions and allowing retransmissions to send duplicate data based on the current packet loss ratio (PLR). The next step was to enforce a time limit for data to be correctly received at the central controller. ENSURE 2.0 was developed by integration of standard network delay formulations, which was effective in providing rapid data delivery within a time frame. However, a small amount of packet losses was detected. To overcome the loss and provide 100% reliability, ENSURE 3.0 was developed to incorporate a forward error correction mechanism. Extensive simulation results are presented to verify the efficacy of the proposed protocols in providing 100% reliability under the given time restraints.

Chapter I

Introduction

Over the last century society has become increasingly dependent on the petrochemical industry. The plants have streamlined the process of cracking petroleum and natural gas to extract and refine ethylene and propylene. Although the majority of society cannot single out the uses of these chemicals, they are sure to use one of the thousands of products they are essential to producing, which range from fuel for our cars to the base compounds of medicine to keep people healthy. While the plants produce essential products, they also endure great risk because the machines that perform the chemical breakdown induce an extremely high explosive risk while surrounded by highly flammable chemicals.

In order to provide a safe environment, sensors are placed throughout the plants to measure the temperature and flow through the pipelines. These readings are then passed through 10/100 Ethernet cables to a programmable logic controller and then on to a distributed control system. Typically, the acreage occupied by a refinery is large making the initial installation of cable very costly. In addition, heavy equipment such as cranes frequently run over the lines leading to damage and requiring additional repair cost. While installing wireless networks would eliminate the high investments in technology, dangers presented by inconsistent transmissions are much higher.

The typical industrial environment contains a high density of metal structures in a geographic area that leads to a high rate for diminishing signal quality. The metallic quality of the material leads to:

1. Reflections, caused by surfaces with dimensions that exceed the wavelength (2.4 GHz is around 3 cm).
2. Diffraction, the creation of secondary waves produced by structures that are impenetrable by wireless transmissions.
3. Scattering occurs when surfaces are of equal or smaller dimensions than the wavelength. The result is that the waves radiate in endless directions.

The cumulative impact of the above events in one environment results in a significant presence of noise in the system which naturally leads to distorted wireless signals and unreadable packets. The standard estimate of signal fading in a petrochemical environment is 20 dB faster than a non-disrupted environment.

The goal of the present protocol ENSURE is to eliminate the need to run obsessive wiring by developing a protocol to overcome the aforementioned challenges to provide 100% reliability for sensor readings through a wireless network. The two factors one must focus on are:

1. Making sure the master node receives and is able to decode each reading.
2. Completion of the task in a timely manner.

ENSURE was developed in three stages, each one aimed at eliminating one negating factor of wireless usage. When developing the first version (i.e. ENSURE 1.0), the most important need was targeted: reliability. From the first transmission, duplications of packets are built-in to combat any possible noise in the environment.

ENSURE 1.0 is a proactive approach toward guaranteed reliability. In the retransmission stage, a scalable window is established, allowing for a real-time adjustments based on current packet loss ratios. With ENSURE 2.0, the focus shifted to time deadlines. Developing more accurate calculations for overall delay means the engineers can determine if a network can provide data within a given time frame (error-free packets should be delivered to the controller station within a certain time window). By enforcing a time restraint, a small number of losses will most likely occur. To correct this ENSURE 3.0 was formulated. More specifically, ENSURE 3.0 deploys forward error correction to increase the likelihood of packet loss recovery at the receiver (e.g., controller base station) to eliminate or reduce the number of retransmissions. This approach suggests that by adding a controlled cyclic error correcting code both reliability and deadlines can be met.

Once all the simulations were completed and the results were graphed, all versions of ENSURE proved to out-perform the widely accepted Reliable Data Transfer 2.0. When comparing the iterations of the new protocol, the findings were consistent with the expected performance. Specifically, ENSURE 1.0 outperformed 2.0 and 3.0 in channel utilization but used a large amount of time. When comparing just ENSURE 2.0 and 3.0, the third version produced the best resource usage and did so in a reasonable amount of time. A detailed analysis is found in Chapter 5 with a comprehensive conclusion detailed in Chapter 6.

The remaining material will be presented in the following format. A detailed look at other research that is being performed in the networking of a petrochemical plant will be seen in Chapter 2. Many recent publications have focused using different types of Hybrid ARQ channels. Each type will be clearly defined and explored. Another concept that will be addressed is the use of forward error correcting to minimized errors.

Chapter 3 will establish the current typology of a typical petrochemical plant as well as the new setup ENSURE aims to make possible. Once this is clear, a walk-through of the development of the new protocol will be presented. Each hurdle to wireless technology in a plant will be addressed step-by-step, and a solution for each will be presented.

Chapter 4 will outline the approach to simulating each version of the protocol. Each design decision that was made will be supported through networking principles. For example, a detailed explanation will be given for using a distributed protocol. In addition, the implantation of the protocol into existing networks will be examined.

Experimental results analyzing the performance of all of the models of ENSURE will be presented in Chapter 5. The results of simulations of versions 1.0, 2.0 and 3.0 running one, two and three sensors will be presented. Also, provided will be a side-by-side comparison of all three protocols along with results that are produced when running the RDT 2.0 protocol.

The final Chapter will be dedicated to forming a decisive conclusion of the effectiveness of the work. Possible adaptations that could be explored in the future will be addressed as well as real world implementation plans.

Chapter II

Related Work

Every day brings the next big invention and with each new product the demand for petroleum based components increases. While consumers push for more production, plants seize the opportunity, often causing oversights in networking maintenance. Unfortunately, by neglecting the sensors and network communications, automated systems are making poor decisions which compromise the safety of workers and the civilians located around the plant. BP's Deepwater Horizon explosion in April 2010 left 11 people dead; this is one of dozens of cases worldwide where implementing reliable sensors would have saved lives and long-term costs. In light of the increase in the number of accidents, researchers have been vigorously working to find improvements that would be accepted as the new sensor standards.

There is a wide range of focus and approaches being explored by researchers. It is important to highlight a few of the main topics receiving wide-spread interest to give proper perspective on the research completed to develop the series of ENSURE protocols.

The most applicable topics of study are the benefits [1]. Sanja Šain provides a detailed analyses of the effects of the main sources of interference ENSURE faces. The effects of reflection, diffraction and scattering are examined and allow protocol designers a clear understanding of the signal deprecation they are working to overcome.

Turning to publications that have the goal of overcoming obstacles, there is great interest in Hybrid-ARQ protocols. When referring to these protocols there are four types that have received interest:

1. Hybrid-ARQ Type I focuses on discarding any packets with errors and future packets are transmitted with error correction information added to the data.
2. Hybrid-ARQ Type II combines new information bits with a limited amount of redundant bits in each packet. The packets that are actual retransmissions contain additional redundant bits that are added to the initial ones. The goal is to give the receiver enough bits to patch all errors in a packet.
3. Hybrid-ARQ Type III aims to provide parity bits within a packet so that even if an error occurs the packet is self-healing.
4. Another approach, called chase, is combining Type III with one of the redundancy measures.

There is an area of research that focuses on lowering redundancy in Hybrid-ARQ Type I. Other researchers focus on enhancing the Hybrid-ARQ Type II by using an iterative bit flip with the use of a turbo coder [2]. While the previous papers worked to improve the Hybrid schemes, many others simply wish to form selection criteria when using the protocols [3], [4]. Different approaches to improving a hybrid scheme are to modify the modulation code [5], [6], [5].

Though first introduced by Richard Hamming in the 1940s, forward error correction (FEC), which attempts to correct errors through pure redundancy, continues to receive a great deal of attention. This method is often used when no feedback

channel is available. In a paper published by Purdue University, a research team presents a way to reduce feedback traffic through the use of coding individual flows and inter-flows of a single hop network [6]. In fact, many approaches to optimizing error correction code are being explored and although the initial test-beds are not wireless sensor networks, when proven effective, they could be tested in this area [7], [8]. Another topic being explored is the best time to detect an error on the network and the most efficient way to correct errors with FEC [9], [10].

The range of interest in FEC varies not only as it relates to analog data sensors, but labs are also using its robust powers for video monitoring as well. In an effort to lower latency with an isolated real time video network, THALES Communications in France studied the effects of Reed Solomon codes [11], [12]. Similar papers have been published in relation to cloud computing and FEC [13] [14]. Also being investigated is the optimal placement of code, i.e. in what layer to incorporate it [15,16], [17], [18], [19].

When working with wireless sensors, many labs have recognized the need to provide reliable networks. Most of the proposed solutions place the majority of their concentration on meeting time restraints while making reliability secondary [20], [21,22,22], [23], [24], [25,26], [27]. Within this field, studies have been done to determine the performance effects of using retransmissions over networks utilizing current IEEE standards [26], [28].

Gaining a clear understanding of wireless channel prediction methods was critical to the development of the ENSURE protocols. Perhaps the most important was to examine loss probability. Since different forms of signal modulation exist, it is important to understand the channel estimation for each. Several papers have been published that outline this factor with the different types of fading channels [29], [30], [31], [32,32],

[33,34], [34]. Working hand-in-hand with these analyses are researchers producing codes to automatically output error-estimation [35], [36], [37], [38]. Often after a new code is developed, researchers begin to use them to analyze other types of codes. As an example, one may ask: Given these new set of criteria, how effective are LDPC codes at increasing performance [39], [40,30]? Tying in to these concepts is finding the ideal way to partially cancel interference sources without disrupting the data source's signal [41], [42], [43].

When deploying a new wireless setup, noise can be avoided by carefully plotting the network topology, therefore studying the effects of sensor placement on channel utilization is in high demand. Interesting connections are being found between angle of arrival, path amplitude and delay [44], [45], [46]. Another application of modeling is finding the maximum channel capacity [47], [48], [49], [50].

Though the work of ENSURE focuses on the use of wireless sensors in petrochemical plants, many other places deploy this technology such as border patrols, geological services, military surveillance, and home health [51], [52], [53]. There are many issues that arise from the use of wireless sensor networks (WSN). Many of the sensors that are in place also have limited access to power; therefore, it becomes essential to find ways to conserve power consumption. One-way to accomplish this is to add on-board intelligence to every unit [54], [55], [56]. Another method of control is to facilitate the savings through scheduling [57,58], [58], [59]. A vast majority of sensor networks are responsible for communicating sensitive information. When they are wirelessly connected, the signals are open to the public unless the transmitted packets are protected. However the topic of security is complex and with each improvement threats become more sophisticated. Constant research has to be done to continue to

protect data. In 2012 alone, a great number of approaches have been presented, each of which is far too complex to minimize to a single sentence [22], [60], [61], [62], [63].

Another major issue in WSNs is providing a tolerable quality of service. With mobile sensors an approach often used is use of object tracking and efficient ways to rendezvous two master nodes [64], [65], [16], [66], [67].

Chapter III

Timely and Guaranteed Packet Transmission

3.1 Introduction

The core problem in placement of wireless technologies in a petrochemical plant is the inability to produce 100% reliability in the presence of a time restraint. In order to clarify the need for this, one must understand the ramifications of the absence of a consistent channel by relating the network to a universal example. Much like communication to and from a war zone, if a message or command is unable to be relayed to either the decision-making unit or to the mechanisms carrying out the orders, the results can be deadly. Society has seen this repeatedly in terms of petrochemical environments over the last decade, with the most memorable example being BP's Deepwater Horizon. However, the general public does not completely understand the way readings are sent back to the computer that controls equipment processes. In this Chapter we will clarify the processes and proceed to examine a method of improvement.

In a war zone, frontline soldiers are responsible for assessing the conditions of the area they are assigned. In an industrial situation, this is the job of the temperature gauges and flow sensors that are placed on the equipment. Each of these taken measurements is sent as packets, let them be called m_s , to an intermediate node, called a program logic controller. The wireless subnetwork used in this phase will be referred to as C. While readings equate to the observations of a soldier, the packets can

be seen as the verbal passing of the message from one level of command to the next. The commanding officer is the equivalent of the base station in this example, while all the people who relay the messages are the PLC units. In the networking world, the medium used to transfer the message is the channel, which can be thought of as person's voice. When there are a lot of other people talking around the person delivering a message, their voices can overpower the speaker's voice. This would be referred to as noise in a network (N). Noise stems from interference, reflection, multipath fading and other environmental factors, which cause the signal energy to drop below a certain level. Consequently, the receiver considers the message as lost (e.g., too distorted to be decoded). Let the likelihood of packet loss can be denoted by p (e.g., $p = 0.5$ suggests that 50% of packets will be lost).

Often the higher ranks of the military have several subordinates reporting to them and expect updates every eight hours. Again, relating this to a wireless network, the number of subordinates the officer has will represent the number of sensors present for the PCL, N_s , and the eight hour reporting cycle would be out delay, D .

Suppose that a message was not received by the commanding officer, he would make efforts to locate his troops, and might even send multiple sources in order to ensure success. By requesting the message be communicated again, he has set up a retransmission window, $rwnd$. In order to determine how many lines of communication the officer tries reach out through, he has to look at all factors in his campaign. For example, did more than one subordinate fail to respond? How long has it been since the last communication? What is the frequency of communication being lost throughout the post?

In a combat situation, the answers to this situation can be intuitive, based on logic or a combination of both. However, in a network there must be clearly devised formulas to find an ideal response. The next few sections will be spent reviewing basic equations and advancing them to develop a real-time solution that guarantees reliability in a timely manner. In the end, an outline of the ENSURE protocols will be clear.

3.2 Defining Basic Equations

All networks, regardless of the medium they use, can be adequately defined in a set of equations that are universally accepted. The challenge presented by petrochemical plants is to provide a wireless transmission protocol that guarantees that all messages are received and properly decoded at the base station. To formulate this protocol there are two key factors that must be considered:

1. Deadline – Referring back to the military example, this is the eight-hour cycle that soldiers have to report back to their commanding officer. In a plant situation, the time frame (D_{TOTAL}) is typically much smaller and is dictated by the equipment the sensors are monitoring.
2. Real-time packet loss ratio – As mentioned in the example, it is possible that more than one subordinate fails to report to the commanding officer, and the count of those failures is calculated into the number of ways the he attempts to establish communication. In the networking sense, the number of readings lost over the lifetime of a channel connection can be calculated to determine the appropriate reaction to a packet loss.

Having established the parameters that are of particular concern to ENSURE, the focus will shift to defining them in mathematically terms. When considering the deadline, one must determine the series of events required to transmit a packet from a sender to

its intended target. First the packet must be pushed from the sender into the network channel; the amount of time this process takes is referred to as D_{TRANS} . This change in time is determined packet size (L), number of packets initially sent ($cwnd$) and the rate of a wireless channel (R) and is written as:

$$D_{TRANS} = \frac{L * cwnd}{R} \quad (1)$$

The next step is propagating the packet across the communication channel and relies on the medium of the transmission channel as well as the distance the sender is from the receiver (d). Since a wireless transmission is mainly sent through the air, the speed it travels is:

$$S = c - 90 \frac{km}{s} \quad (2)$$

Which leads to:

$$D_{PROP} = \frac{d}{S} \quad (3)$$

When acknowledgement is expected, the values of D_{TRANS} and D_{PROP} are doubled, thus the total time required for a transmission is:

$$D_{TOTAL} = 2 (D_{TRANS} + D_{PROP}) \quad (4)$$

The other widely accepted formula that ENSURE will use helps to determine the packet loss ratio (PLR) of the link. By definition a ratio is a comparison of two numbers, in this case those are the number of losses experienced in the network and the total number of packets sent (TX).

$$PLR = \frac{p}{TX} \quad (5)$$

Since both the denominator and numerator are based on packets coming from the same source, you can never lose more than you transmit, therefore, $PLR \leq 1$.

$$PLR \leq 1 \quad (6)$$

3.3 Addressing Reliability

The foundation of ENSURE is based on a binary erasure channel (BEC), meaning that the channel can only receive the packet or a loss. The probability a loss occurs is independent of the transmissions received before or after it. Consequently, the likelihood of packet loss is equivalent to the observed packet loss ratio (PLR). Therefore, the capacity of a binary erasure wireless channel C is:

$$C = 1 - PLR \quad (7)$$

This equation suggests that the optimal throughput that can be achieved over such channel is determined by cost PLR. As a result, the ideal performance for any reliable transfer protocol (including ENSURE) cannot exceed C .

Let β denote the operation cost coefficient, which is defined as follows: The percentage of the channel used to send a unique packet during retransmission

$$\hat{C} = \beta \times C \quad (8)$$

For convenience, the following table provides a visual of what β would be under a variety of circumstances and also indicates the number of packets that ENSURE would send.

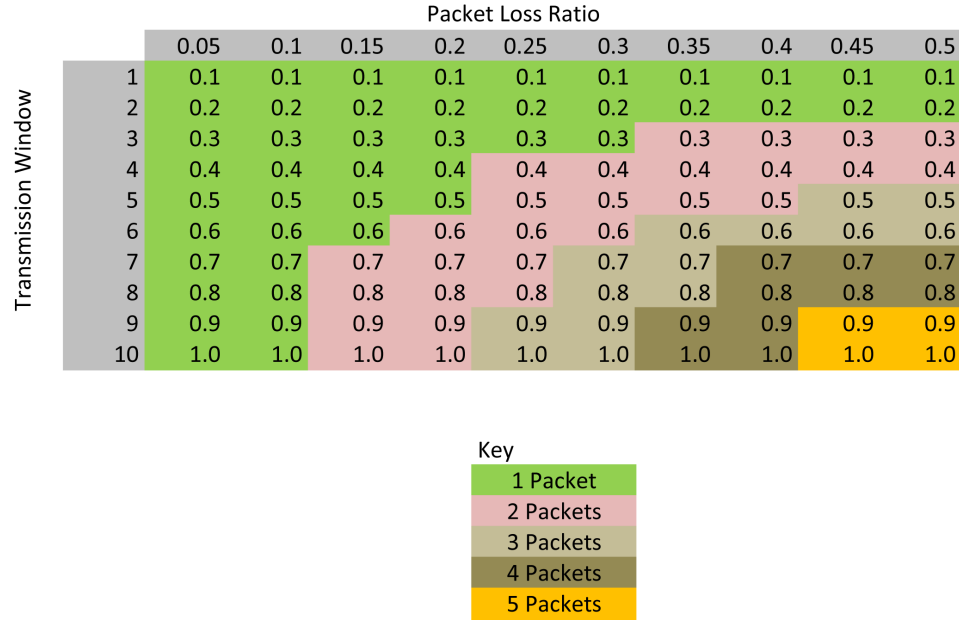


Figure 1. Operation cost coefficient compared to number of required retransmissions

Having established the probability of a correctly received packet, the goals of ENSURE need to be revisited; namely that of 100% reliability. For a channel to be completely reliable, every unique packet, or sensor reading in this case, will reach the base station irrelevant of any retransmissions needed. In fact a wireless network will lose packets; it is impossible to have a perfect channel. This leads to a critical question that ENSURE must answer: how will it recover a loss packet?

In an ideal situation, readings would be decodable after the first round of transmissions. In an average industrial environment noise levels routinely reach 50%, meaning that initially sending just one packet per reading will result in a high loss level. An adjustment to the size of the first transmission round will allow ENSURE to combat the known *PLR* right away. When no working ratio has been established the protocol

assumes $PLR = .5$. Each sensor will be sending one unique reading per frame; this reading will be duplicated k times. The variable will be determined by the following equation:

$$k = \text{ceil}\left(\frac{1}{PLR}\right) \quad (9)$$

Establishing the

$$cwnd = k * N_s \quad (10)$$

where N_s is the number of sensors.

Determining the number of packets lost in transmission is as simple as subtracting the number of correctly decoded unique packets from the expected number. By dividing the number of losses by the probability that a packet was correctly received, one can calculate the size of the retransmission window that will overcome the noise of the environment.

$$rwnd = \text{ceil}\left(\frac{p}{1-PLR}\right) \quad (11)$$

The process of calculating a $rwnd$ can be repeated until the base station correctly decodes a packet for every reading taken by the sensors, forming the enter inner workings of ENSURE 1.0.

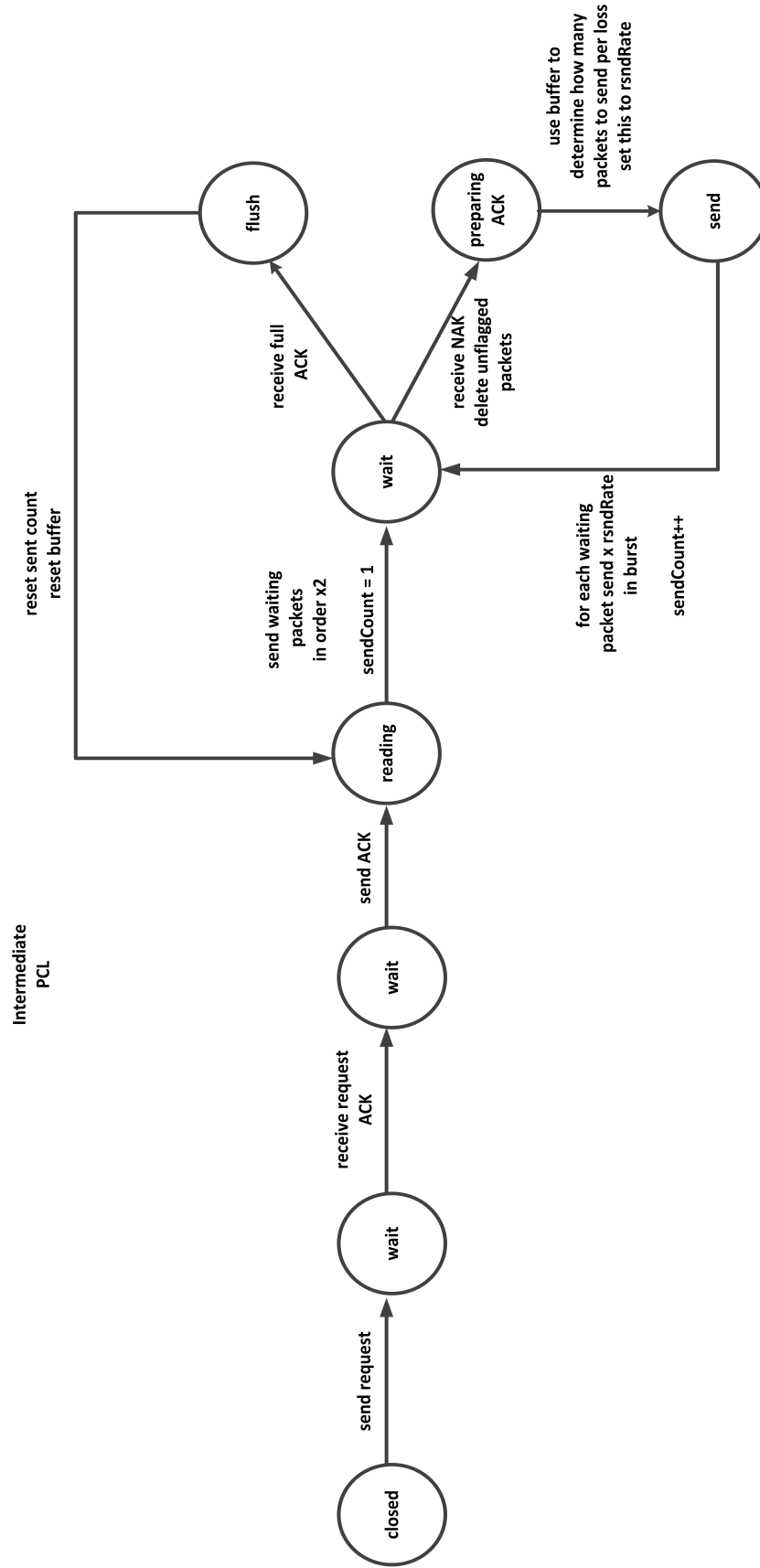


Figure 2. Finite state machine for ENSURE 1.0 – intermediate node

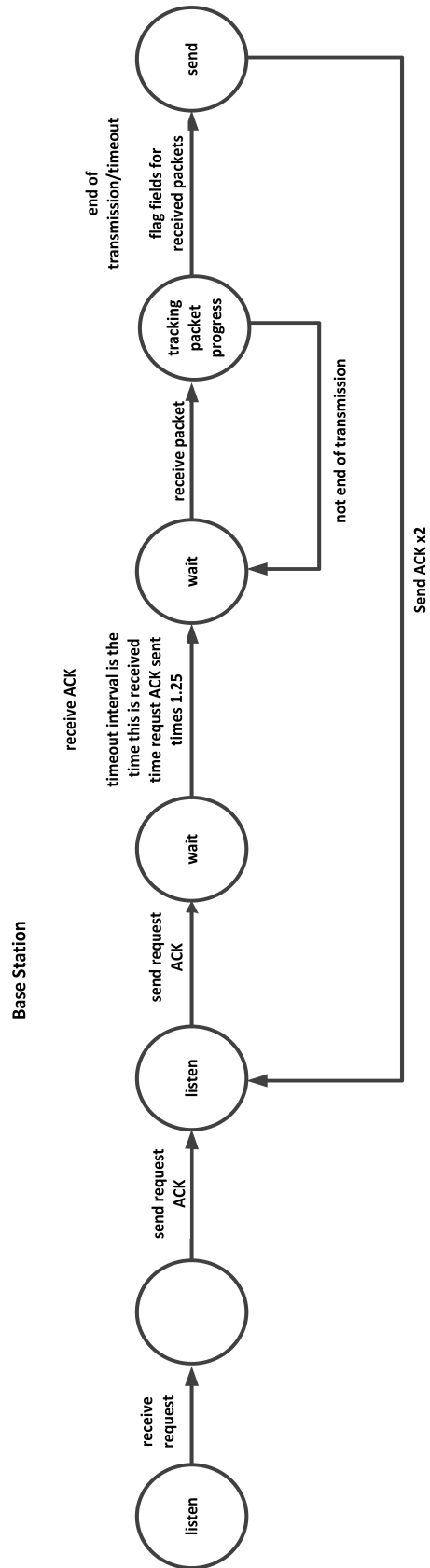


Figure 3. Finite state machine for ENSURE 1.0 – base station

Ensure 1.0

<i>Intermediate PCL</i>	<i>Base Station</i>
loop(forever){	loop(forever){
switch (event)	event: receive packet
event: data collected from all subnetworks	attempt to read;
start timer;	if (able to read)
send each reading twice;	mark sensor reading as received;
sendCounter = 1;	if (last transmission in frame)
break;	send NACK indicating packets;
event: NACK received	else
if (one or more packets were lost)	wait;
delete all ACKed packets;	event: timeout
determine number of times to resend	send NACK indicating received packets
NACKed readings;	}
send;	
sendCount++;	
break;	
else	
empty retransmission queue;	
break;	
event: timeout	
resend queue;	
break;	
}	

Figure 4. Algorithm for ENSURE 1.0

3.4 Meeting a Deadline

Now a way to provide reliability has been formulated, a re-examination of the needs of the petrochemical industry provides a second hurdle that ENSURE needs to overcome: a restricted time frame. Version 1.0 was allowed to run until a reading was correctly received and could result in extremely high values of D_{TOTAL} . To enforce a protocol time limit one must set D_{TOTAL} equal to our maximum allowable value. However, the equation provided to this point, Equation 4, is written in generic terms and must now be adapted to meet the specifications of ENSURE.

Since the size of the transmission window adjusts with each round of retransmissions, the value of D_{TRANS} is no longer static. Therefore, Equation 1 will have to be modified each time a new request for retransmissions is sent. Here the maximum number of transmissions will be set to two, the initial transmission and one retransmission round, so a clear explanation of the equations can be presented. In this

spirit new variables can be defined for delays, one for the initial transmission, D_{ITRANS} , and one for the retransmission round, D_{RTRANS} .

$$D_{ITRANS} = \frac{L * cwnd}{R} \quad (12)$$

$$D_{RTRANS} = \frac{L * rwnd}{R} \quad (13)$$

Now that these two values are established, a new equation for D_{TOTAL} can be formulated. Going back to the first equation for this variable, Equation 4, the total delay of the network simply doubled the amount of time a one-way transmission took. However, since there are now different values for transmission delays, this has to be reformulated to:

$$D_{TOTAL} = D_{ITRANS} + D_{RTRANS} + 2(D_{PROP}) \quad (14)$$

Depending on the parameters given to an engineer, he can now solve this equation for two things: maximum amount of tolerable delay or the number of sensors a given set-up can handle. Having incorporated these two things, ENSURE 2.0 was born. The predicate outcome of the new protocol greatly reduces values of D_{TOTAL} , but with a small number of losses or erasures.

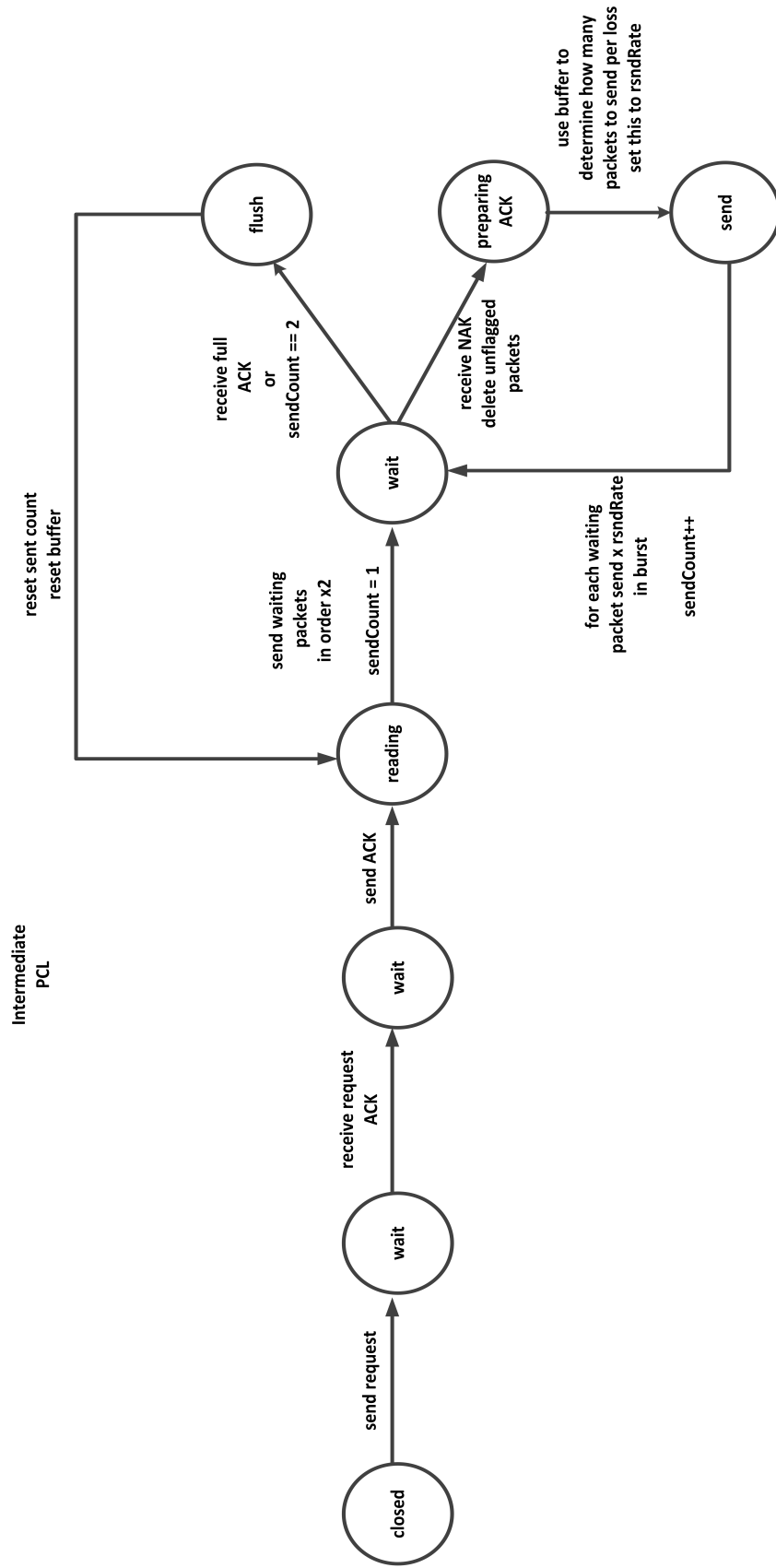


Figure 5. Finite state machine for ENSURE 2.0 – intermediate node

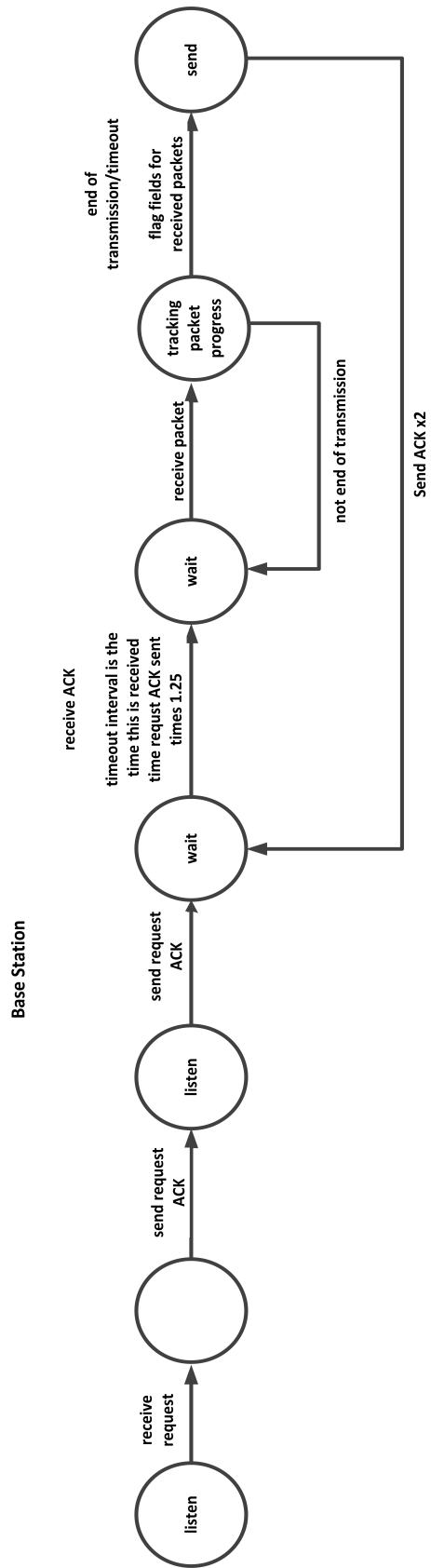


Figure 6. Finite state machine for ENSURE 2.0 – base station

Ensure 2.0

<i>Intermediate PCL</i>	<i>Base Station</i>
<pre>loop(forever){ switch (event) event: data collected from all subnetworks start timer; send each reading twice; sendCounter = 1; break; event: NACK received if (one or more packets were lost && sendCount == 2) delete all ACKed packets; determine number of times to resend NACKed readings; send; sendCounter++; break; else empty retransmission queue; break; event: timeout resend queue; break; }</pre>	<pre>loop(forever){ event: receive packet attempt to read; if (able to read) mark sensor reading as received; if (last transmission in frame) send NACK indicating packets; else wait; event: timeout send NACK indicating received packets }</pre>

Figure 7. Algorithm for ENSURE 2.0

3.5 The Best of Both Worlds

ENSURE 1.0 accomplished reliability, while 2.0 overcame a time restraint, but given only these options one must choose either complete reliability or time utilization. The natural next step is to devise a way to combine the two. Given the predicted number of complete losses from ENSURE 2.0 are relatively low; a natural progression is to a cyclic error-correcting code that will give the network the ability to recover some of the erasures.

Reed Solomon, often used in coding theory [68], finds the most popular subset of binomial data in a packet and periodically sends a packet dedicated to this subset (a parity packet). The downside to this is by transmitting a parity packet you lose the opportunity to send actual data, but this one packet has the potential of containing patch data for multiple packets. When a network is sure to encounter losses the net gain makes the use of the resources well justified.

Placing an encoder before the packet transmission and a decoder after the channel in ENSURE 2.0 forms ENSURE 3.0. The expected outcome is that the network will be able to completely recover any losses, thus providing guaranteed reliability even when implementing a time restraint.

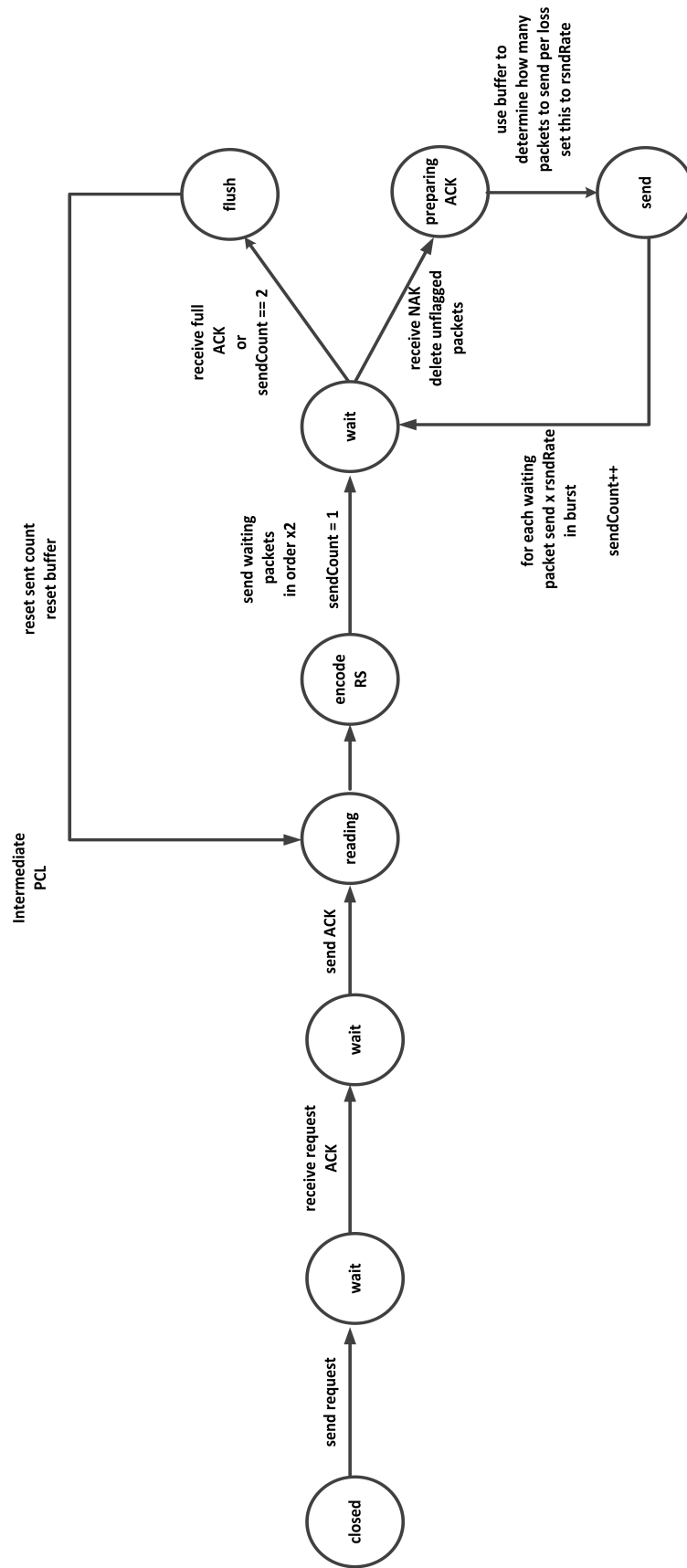


Figure 8. Finite state machine for ENSURE 3.0 – intermediate node

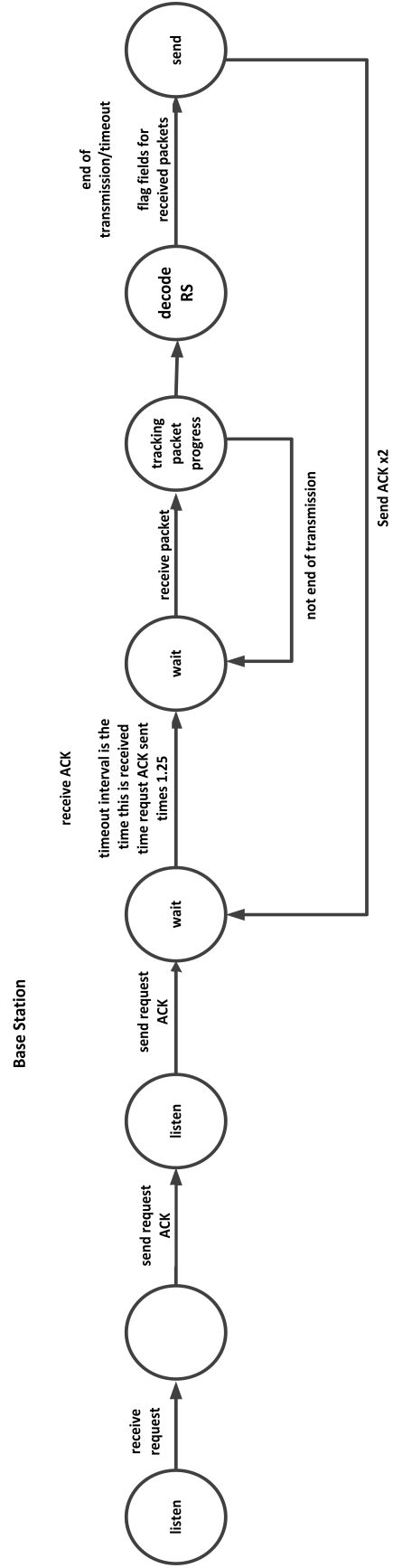


Figure 9. Finite state machine for ENSURE 3.0 - base station

Ensure 3.0

<i>Intermediate PCL</i>	<i>Base Station</i>
loop(forever){	loop(forever){
switch (event)	event: receive packet
event: data collected from all subnetworks	decode with Reed Solomon
encode with Reed Solomon	attempt to read;
start timer;	if (able to read)
send each reading twice;	mark sensor reading as received;
sendCounter = 1;	if (last transmission in frame)
break;	send NACK indicating packets;
event: NACK received	else
if (one or more packets were lost	wait;
&& sendCount == 2)	event: timeout
delete all ACKed packets;	send NACK indicating received packets
determine number of times to resend	}
NACKed readings;	
send;	
sendCount++;	
break;	
else	
empty retransmission queue;	
break;	
event: timeout	
resend queue;	
break;	
}	

Figure 10. Algorithm for ENSURE 3.0

Chapter IV

Approach

4.1 Refining Typology

When designing a network, different factors including the type of information that will be transmitted and the kinds of equipment must be considered. Industrial environments make use of Supervisory Control and Data Acquisition (SCADA) management systems. The way this type of environment works is that the sensors take readings on the pipeline level and pass them through a PLC on up to a distributed control system.

Some plants currently have short-range wireless networks that connect the sensors to the PLC. The length from the sensor's wireless transmitter to the receiver is kept to a minimum and is only utilized when running a wire to the sensor would be almost impossible. One example is a sensor that measures the temperature at the top of a smoke stack, which is very tall. Two key areas in a refinery are the pipeline and the distillation unit. Figure 8 shows how a petrochemical plant that uses wireless technology would be configured if deployed with currently accepted standards. On the sample pipeline, two temperature gauges are connected to a PLC dedicated to ensure the readings get back to the control unit. Similarly, a sample distillation unit has four pipes whose temperature must be monitored. While the sensors are connected wirelessly to their respective PLC, 10/100 cables are installed from each PLC to the distributed control unit. Each cable can be hundreds of yards long and extremely costly to deploy and maintain. However, with current standards, moving toward a completely wireless

system is virtually impossible due to the types of equipment that is monitored (a detailed explanation can be found in Chapter 1).

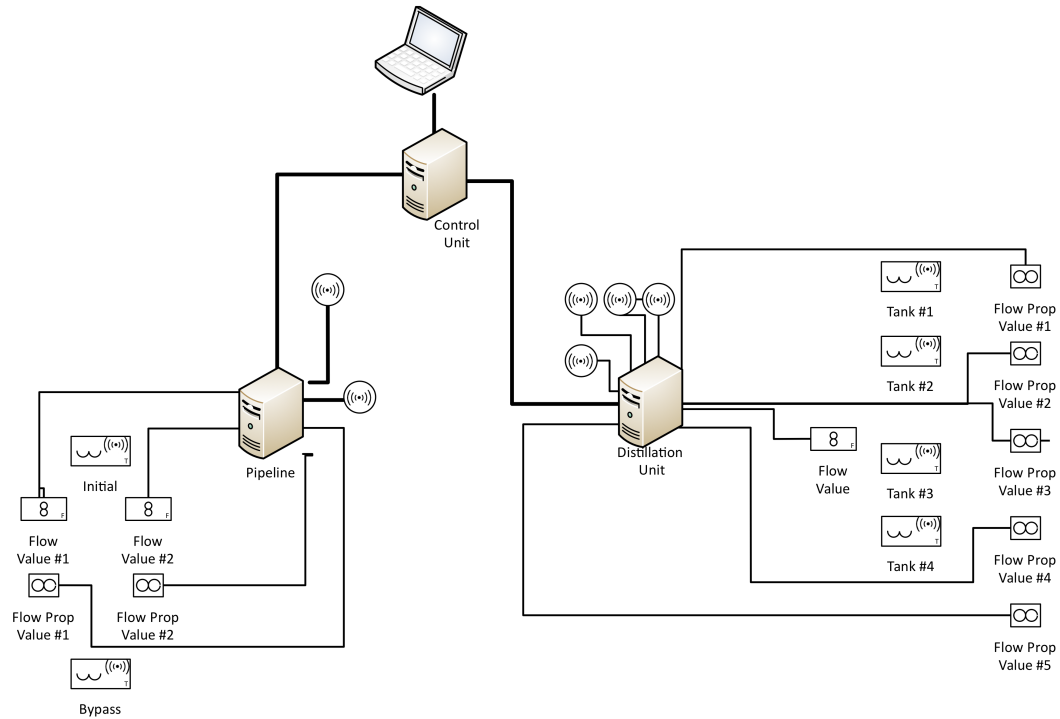


Figure 11. Sample of network topology currently used

A primary goal of all forms of business is to minimize operating costs; deploying a completely wireless sensor network will be cost beneficial for petrochemical plants from the following aspects. (1) The initial setup cost will be lower, (2) the cost of paying for repair materials and getting a qualified technician to perform them will be less, and (3) downtime of the plant will be reduced leading to an increase in production.

Developing a new typology is a key step in lowering operating cost. When ENSURE is implemented the goal is to refine the wired connection from all the PLCs to the control system to be wireless (shown in Figure 9).

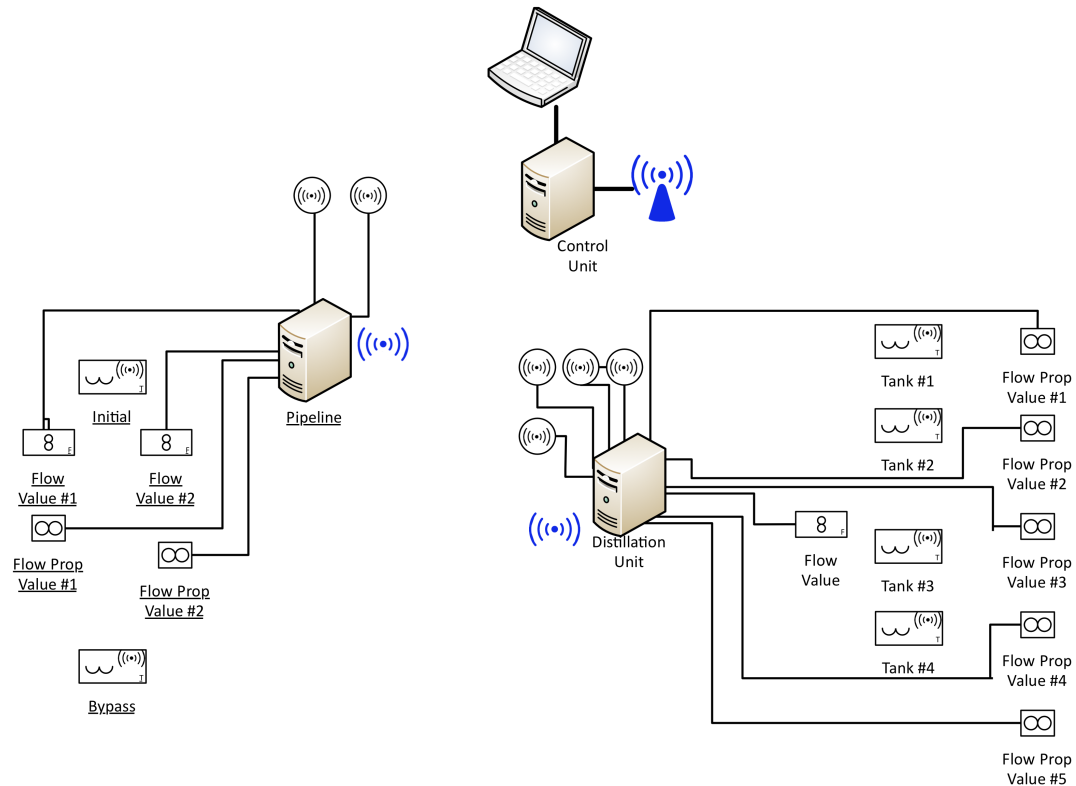


Figure 12. Sensor typology with ENSURE implemented

4.2 Determining Network Connections

The next factor that has to be considered is signal management, i.e. would the wireless sensors still make use of the intermediate nodes? In short, yes, after investigating the options the use of PLCs were found to be beneficial to the network. To further explain, there are two types of networks: isolated and non-isolated. An isolated network exists when one-way communication is established, commonly referred to as a broadcast channel. A non-isolated channel allows for both the sender and the receiver to communicate with each other, a classic example is an Internet connection.

The majority of available SCADA sensors on the market are dummy sensors; they are comprised of a reading mechanism, a transmitter and a wireless antenna.

There is not an onboard processor that would allow it to receive any signals. By definition any wireless transmissions from the sensor will be isolated. Going back to the goal of creating a 100% reliable wireless network, without a method to explicitly request a reading to be resent the reliability requirement cannot be guaranteed.

On the other hand, programmable logic controls have the capabilities to both send and receive. Therefore, a non-isolated network can be configured between each PLC and the control system. Combining the two networks opens up an opportunity to have met the stated needs.

4.3 Protocol Placement

All new technologies have to conquer initial resistance by the consumer to use them. If there is a high cost to enter the market then the consumer is less likely to consider the technology as an alternative. By modifying existing technologies to accommodate changes, the adoption rate would be higher.

A computer network consists of five layers: (1) application, (2) transport, (3) network, (4) data link and (5) physical layers. Carefully positioning the logic of ENSURE within the five layers (presented in Chapter 3) will allow deployment using existing hardware. The application layer is responsible for establishing a client to server connection while the transport layer handles all the communicating factors in the connection. Examples of services provided in the transport level are data modulation, multiplexing, flow control and reliability. Adding ENSURE as an adapter between the application and transport layers will allow the protocol to strictly focus on accomplishing reliability.

4.4 Simulation Models

Three software packages were considered: NS2, OMNet++, and MATLAB with Simulink. After weighing the pros and cons of each package, MATLAB with Simulink was chosen. A main consideration was the availability of the software to The University of Texas at Tyler making it easier for future researchers to expand from ENSURE. For each version of ENSURE a simulation model was developed.

4.4.1 ENSURE 1.0

Before a model of ENSURE 1.0 could be built, a basic wireless simulation had to be in place. Simulink gives engineers the ability to use blocks to map out the path of the network. The network that was built used an AWGN channel to simulate the errors.

From this point, a replica of the proposed typology was developed. One sensor was added and connected to the PLC through an isolated channel, creating one subnetwork. Once the PLC received the readings they were passed to a second subnetwork. Before a packet left the PLC through the second subnetwork, it was processed using the PLC ENSURE logic. Following processing, all packets were sent through a separate channel to the control unit. There, the controller simulated its given logic and accordingly a NACK was sent back to the PLC. Once the NACK was received, retransmissions for missing packets were sent.

In ENSURE 1.0, the network allowed a unique packet to go through the process as many times as needed until it was received error free. Each simulation represented a 50,000 second run, and at the end channel conditions were recorded. Changing the probability of an error in the AWGN channel induced noise levels, and final channel conditions were recorded for a noise range of 0 to 50% in 5% increments. This set of simulations were run when subnetwork one contained one, two and three sensors.

4.4.2 ENSURE 2.0

To test ENSURE 2.0, a way to place a delay limit on the model had to be found. The chosen method was to only allow a unique reading to go through the retransmission process for a predetermined number of times. Similar to the way the equations for this process were explained, the PLC was only allowed to cycle through the retransmission logic once.

The goal of this change was to simulate a deadline for the base station to receive the readings (before the deadline) and make adjustments to the plant processes. Also, the results would provide another variable for engineers to consider when implementing the networks in the field. The same simulation parameters were used to measure the effects on channel conditions using this logic.

4.4.3 ENSURE 3.0

While the previous two protocol simulations address the individual goals of the protocol, complete reliability (ENSURE 1.0) and meeting time restraints (ENSURE 2.0), they do not satisfy the overall purpose.

To further investigate the feasibility in ensuring 100% reliability within a given end-to-end deadline, ENSURE 3.0 is proposed to plug in the channel conditions found for each of the simulations of ENSURE 2.0 and allow Reed Solomon to attempt to correct erasures (MATLAB has Reed Solomon functions as a part of the communication tool box, to simulate ENSURE 3.0 these functions were used). Since parity packets are used, the adjustments to packet loss, unique packets sent and channel utilization were noted.

Chapter V

Experimental Results

5.1 Introduction

The Transmission Control Protocol (TCP) places emphasis on congestion control and maximum bandwidth utilization while making a good faith effort to provide reliability. The petrochemical industry requires 100% reliability from sensors in metallic environments, which results in highly diminished signal quality. ENSURE 1.0 guarantees that the readings taken at the sensor level make it to the base station of the plant, regardless of how much time lapses.

A model of the proposed protocol was made in Simulink and combined with MATLAB to produce a simulation for multiple environments. Simulations of noise levels from 0% to 50% in steps of 5% were run, and the model recorded the corresponding delay, channel utilization, reliability and total losses. When setting up the sensor typology of a plant all pros and cons of potential layouts must be considered. In that spirit there are several key measures which should be analyzed to obtain a complete picture of each protocol.

5.2 ENSURE 1.0

ENSURE 1.0 guarantees that the readings taken at the sensor level make it to the base station of the plant, regardless of how much time lapses. This is accomplished

through use of a retransmission window that adjusts to the current reliability of the channel.

5.2.1 Delay versus Channel Utilization

When determining the amount of tolerable delay in a network, the amount of channel utilization gained through the delay needs to be considered. Another important aspect of the measure is determining if more sources of information can be added to fully capitalize on the available bandwidth in a network.

When implementing one sensor the amount of channel utilization ranges from 30% to 40%, as noted in Figure 10. The usage remains fairly steady throughout the simulation. With two sensors, utilization levels range from 45% to 75% but the bulk of the samples achieved is 60% to 70%, as seen in Figure 11. At 1100 seconds and beyond of delay ENSURE 1.0 steadily approaches an optimal channel utilization of 70%. Three sensors take full advantage of the channel, ranging from 90% to 100% no matter what the delay (Figure 12). This can be attributed to the burst retransmission format of the protocol. By using a *cumulative distribution function* graph, a clear comparison of the network performance of this protocol can be made (Figure 13). The results show that the greater the number of sensors connected to the PLC the more networking resources are used to their maximum potential.

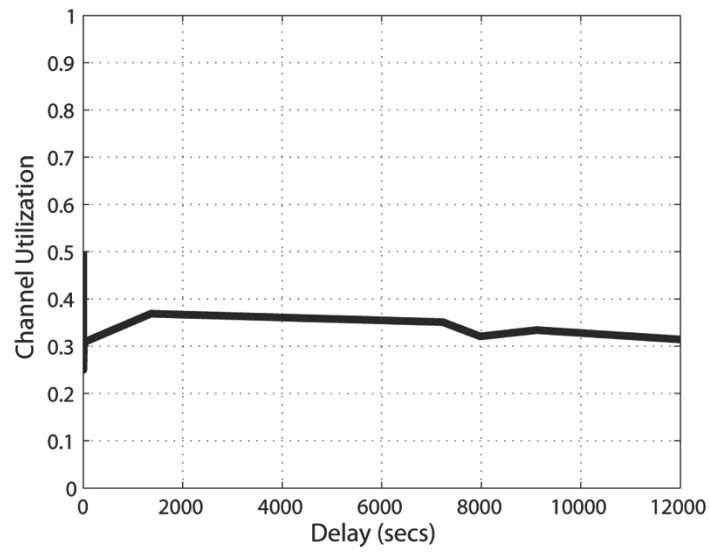


Figure 13. Delay versus channel utilization for one sensor using ENSURE 1.0

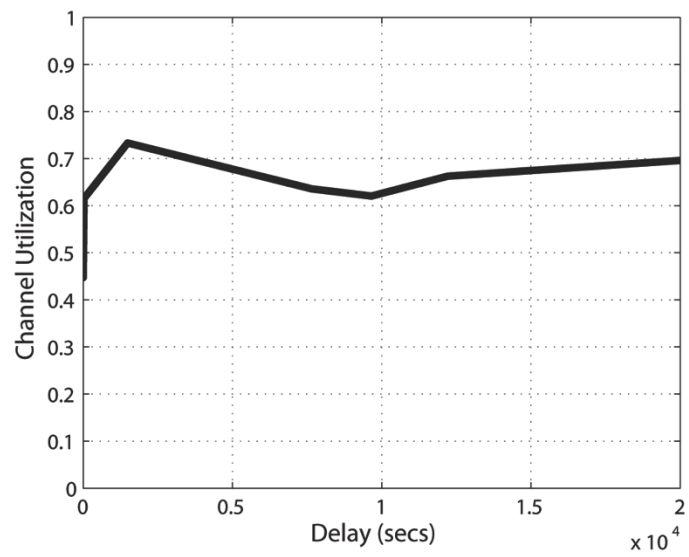


Figure 14. Delay versus channel utilization for two sensors using ENSURE 1.0

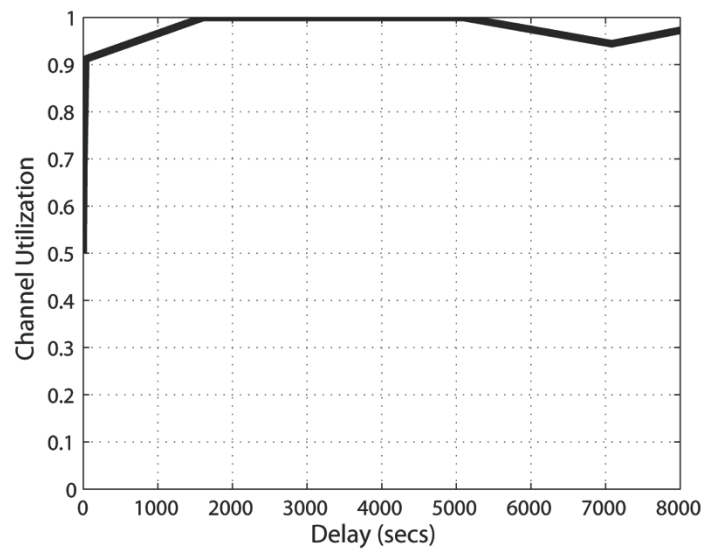


Figure 15. Delay versus channel utilization for three sensors using ENSURE 1.0

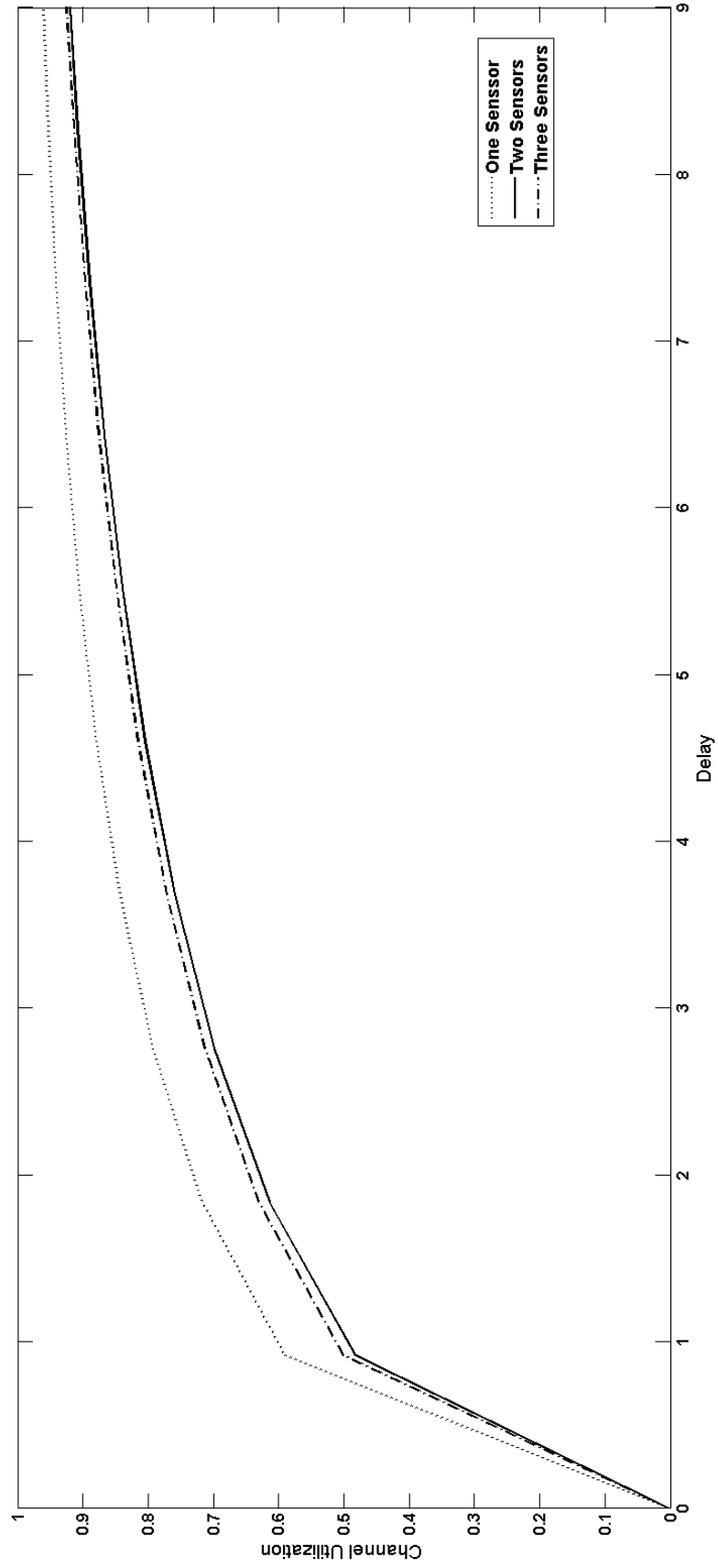


Figure 16. Comparing delay versus channel utilization for all sensors using ENSURE 1.0

5.2.2 Noise versus Channel Utilization

In a perfect world, wireless networks could be isolated from interference during transmissions. Unfortunately, this is impossible because every environment experiences some level of noise. Typically, the higher the noise level the lower the channel utilization level. ENSURE 1.0 was designed to react to the packet losses by adjusting the number of burst transmissions for each packet loss. However, sending these retransmissions also decreases the channel utilization because each duplicate transmission costs the network in resources. By examining the noise versus channel utilization graphs, further understanding of the protocol is gained. Figure 14 shows some interesting effects that the adjustments can have on a network with one source of readings. The network starts at a maximum channel utilization of 50% with no noise then steadily decreases until you have 20% noise, at 30% it recovers to around 40%. This can be explained by the nature of the simulation. The formulas used to determine the number of retransmissions per packets lost is based on the current packet loss ratio of the network and are always rounded up when a fraction is calculated. Intuitively, the noise in the environment causes more losses, leading to a higher PLR and this leads to sending more packets per loss. The reason there is a dramatic increase between noise levels of 20% and 30% is at the 20% mark the formulas instruct the network to send less packets per loss but this process must be repeated. At the 30% mark, it sends more packets per loss and the base station is able to correctly receive the readings in fewer rounds of transmissions. The same is true when an additional sensor is added (as shown in figures 15 and 16), but at the points the extra transmissions occur differ because the window size increases placing more packets into the network at all times. More packets in a transmission result in the PLR being adjusted in an exponential format. In this measure it is again noted that the more sensors added to the PLC's responsibility the greater the channel utilization (Figure 17).

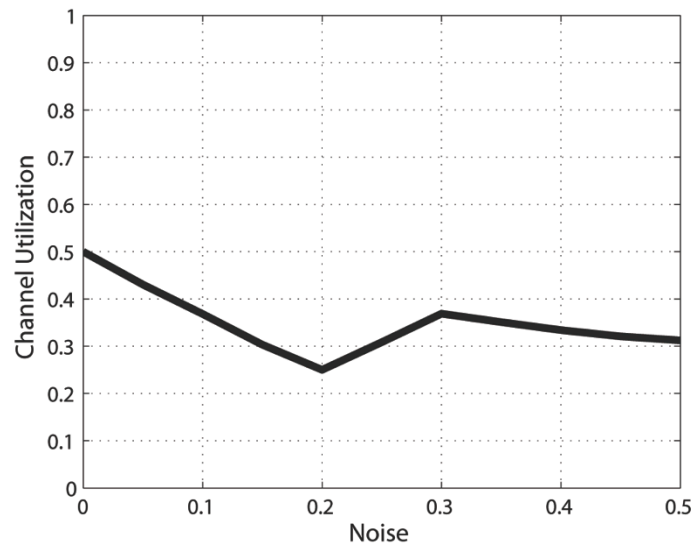


Figure 17. Noise versus channel utilization for one sensor using ENSURE 1.0

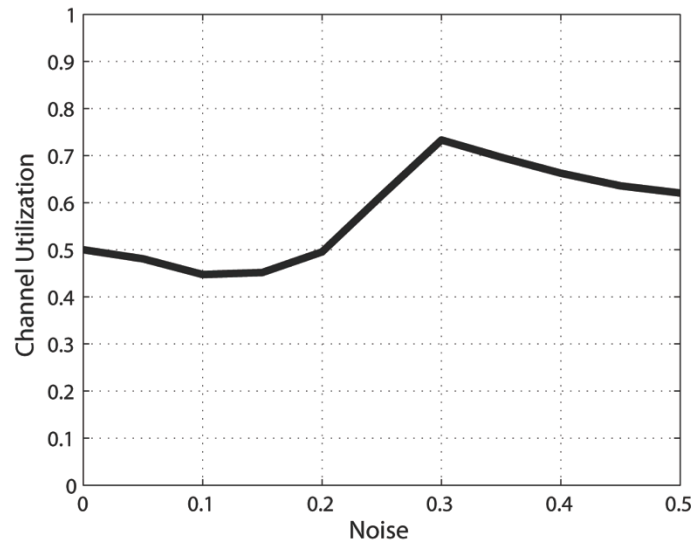


Figure 18. Noise versus channel utilization for two sensors using ENSURE 1.0

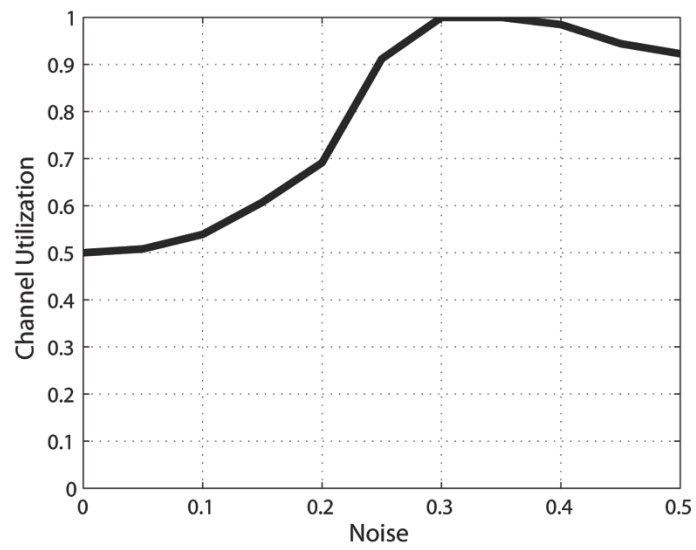


Figure 19. Noise versus channel utilization for three sensors using ENSURE 1.0

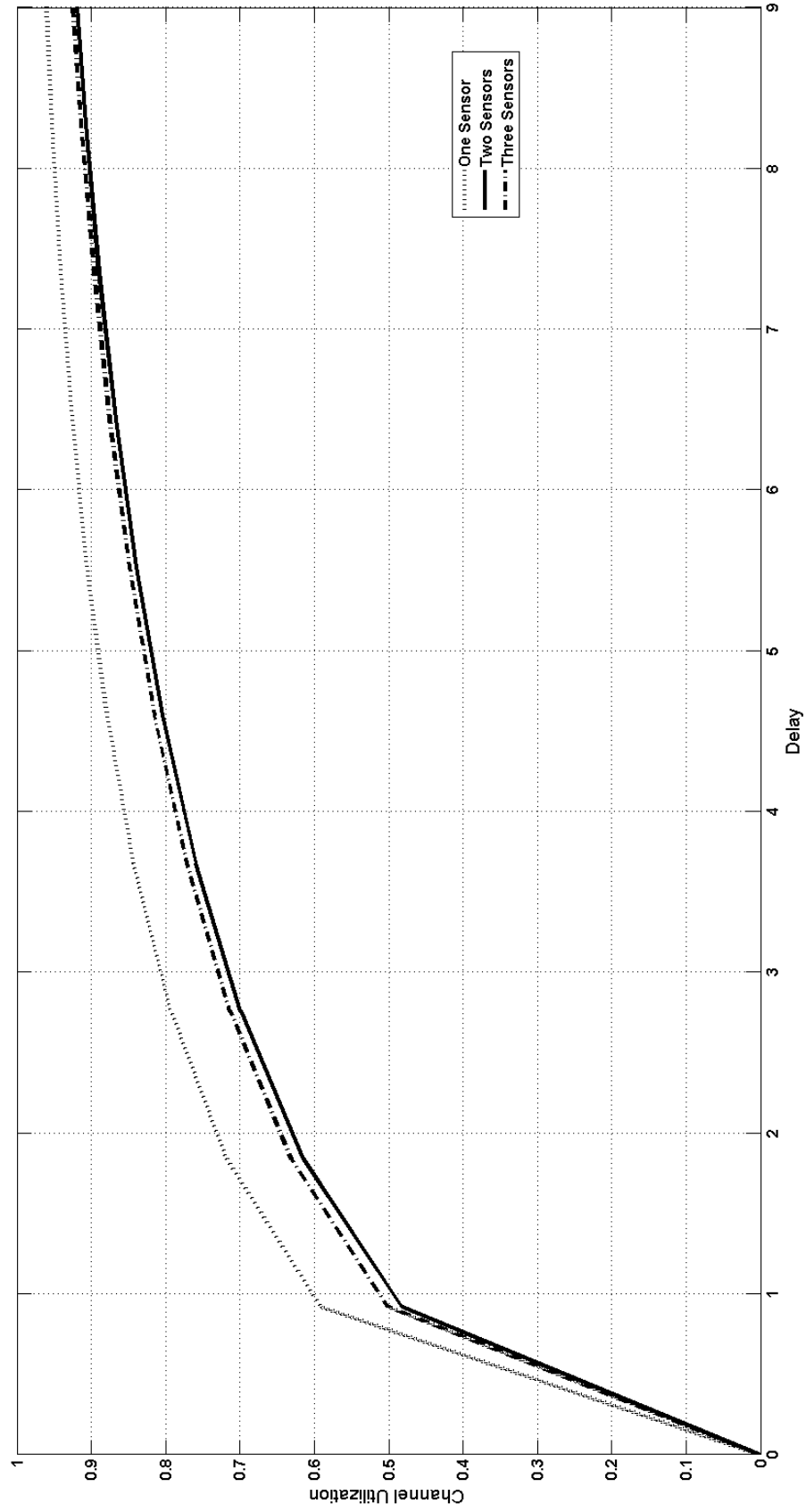


Figure 20. Noise versus channel utilization for all simulations using ENSURE 1.0

5.2.3 Channel Utilization versus Reliability

ENSURE 1.0 was designed in an effort to provide guaranteed reliability no matter what the noise level. Tables 1, 2, and 3 serve to prove the protocol meets this expectation. Let it be noted that a reliability of one indicates that absolutely no packets were lost over the course of the simulation. This measure proves that the protocol works and 100% reliability can be provided when there are no time restraints on the delay.

Table 1. Channel utilization versus reliability for one sensor using ENSURE 1.0

One Sensor		
Channel Utilization	Noise	Reliability
0.25	0.20	1
0.30	0.15	1
0.31	0.25	1
0.31	0.50	1
0.32	0.45	1
0.33	0.40	1
0.35	0.35	1
0.37	0.10	1
0.37	0.30	1
0.43	0.05	1
0.50	0.00	1

Table 2. Channel utilization versus reliability for two sensors using ENSURE 1.0

Two Sensor		
Channel Utilization	Noise	Reliability
0.45	0.10	1
0.45	0.15	1
0.48	0.05	1
0.50	0.20	1
0.50	0.00	1
0.62	0.25	1
0.62	0.50	1
0.64	0.45	1
0.66	0.40	1
0.70	0.35	1
0.73	0.30	1

Table 3. Channel utilization versus reliability for three sensors using ENSURE 1.0

Three Sensor		
Channel Utilization	Noise	Reliability
0.50	0.00	1
0.51	0.05	1
0.54	0.10	1
0.61	0.15	1
0.69	0.20	1
0.91	0.25	1
0.92	0.50	1
0.94	0.45	1
0.98	0.40	1
1.00	0.30	1
1.00	0.35	1

5.2.4 Noise versus Retransmissions

The number of retransmissions the network requires to successfully decode a reading is dependent on the amount of noise in the environment. This is due to the

increased number of losses and that protocol calls for more duplications at higher packet loss levels. In all the ENSURE 1.0 simulations, it was found that between 0% and 20% the amount of retransmissions required was relatively low, between 35% and 50% that number became drastically high. Again, this is consistent when running one, two or three sensors, but the amount of increase at each noise level varies. For example, one sensor has a steady increase (Figure 18), while the simulation for three sensors increased more between 25% and 30% (Figure 20), and then slowed at 30% to 35%. These results tie in to the total network delay to help determine optimal setups, however, by themselves cannot determine a setup.

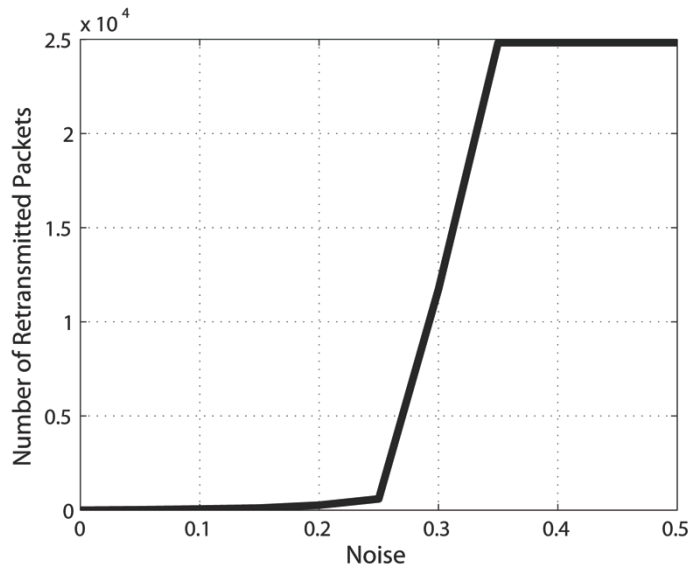


Figure 21. Noise versus retransmissions for one sensor when using ENSURE 1.0

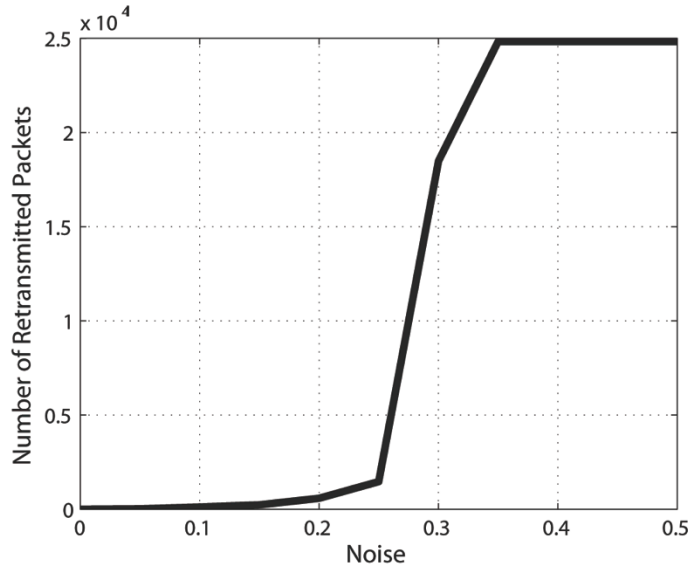


Figure 22. Noise versus retransmissions for two sensors when using ENSURE 1.0

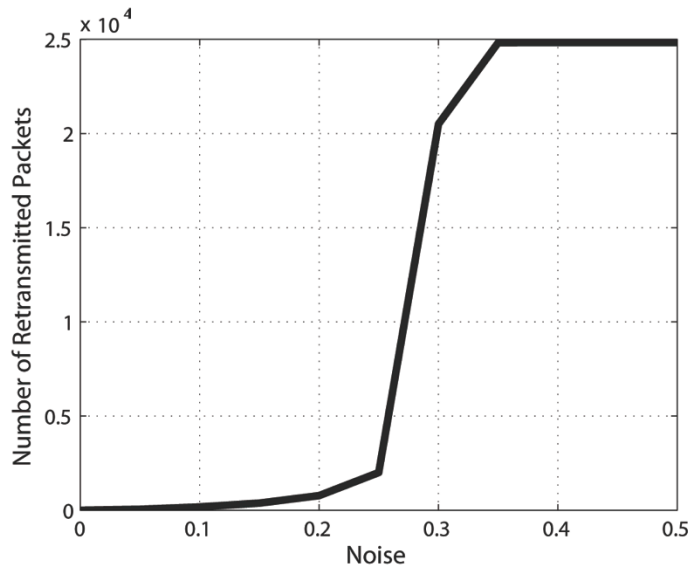


Figure 23. Noise versus retransmissions for three sensors when using ENSURE 1.0

5.2.5 Reliability versus Retransmissions

The main goal of ENSURE 1.0 was to eliminate packet loss through repeat transmissions. Tables 4, 5, and 6 show that the algorithm used accomplishes the goal. Also, we notice that the amount of retransmissions at 50% noise remains steady, but the rate that the protocol increases to this number is proportional to the number of sensors

on the network. Since in the Simulink environment a time limit to run the model must be used, the amount of retransmission reaches a maximum of this constraint. These results can provide insight to the industry because the number of transmissions going between the PLC and the base network will add further noise to the network.

Table 4. Reliability versus retransmissions for one sensor using ENSURE 1.0

One Sensor		
Retransmissions	Reliability	Noise
0	1	0
24	1	0.05
58	1	0.1
112	1	0.15
265	1	0.2
597	1	0.25
11731	1	0.3
24825	1	0.35
24825	1	0.4
24825	1	0.45
24825	1	0.5

Table 5. Reliability versus retransmissions for two sensors using ENSURE 1.0

Two Sensors		
Retransmissions	Reliability	Noise
0	1	0
24	1	0.05
119	1	0.1
235	1	0.15
578	1	0.2
1461	1	0.25
18479	1	0.3
24825	1	0.35
24825	1	0.4
24825	1	0.45
24825	1	0.5

Table 6. Reliability versus retransmissions for three sensors using ENSURE 1.0

Three Sensors		
Retransmissions	Reliability	Noise
0	1	0
53	1	0.05
174	1	0.1
375	1	0.15
774	1	0.2
2001	1	0.25
20504	1	0.3
24821	1	0.35
24825	1	0.4
24825	1	0.45
24825	1	0.5

5.2.6 Noise versus Packet Loss

In every network it is important to look at the network conditions from all angles, and while showing the noise versus packet loss may seem repetitive, it is important to provide. Again, no packets are lost when noise or packet sources are added to the network.

5.2.7 Noise versus Domain of Delay and Channel Utilization

The final measurements were derived by normalizing both delay and channel utilization into a domain and the plotting it against noise. This process helps to identify clusters within the data. The ideal delay points are located in the lower left hand corner while channel utilization points should be in the upper right corner. Figures 21 – 23 all show a natural cluster in the upper right hand corner. This indicates that at high delay times channel utilization and noise are high.

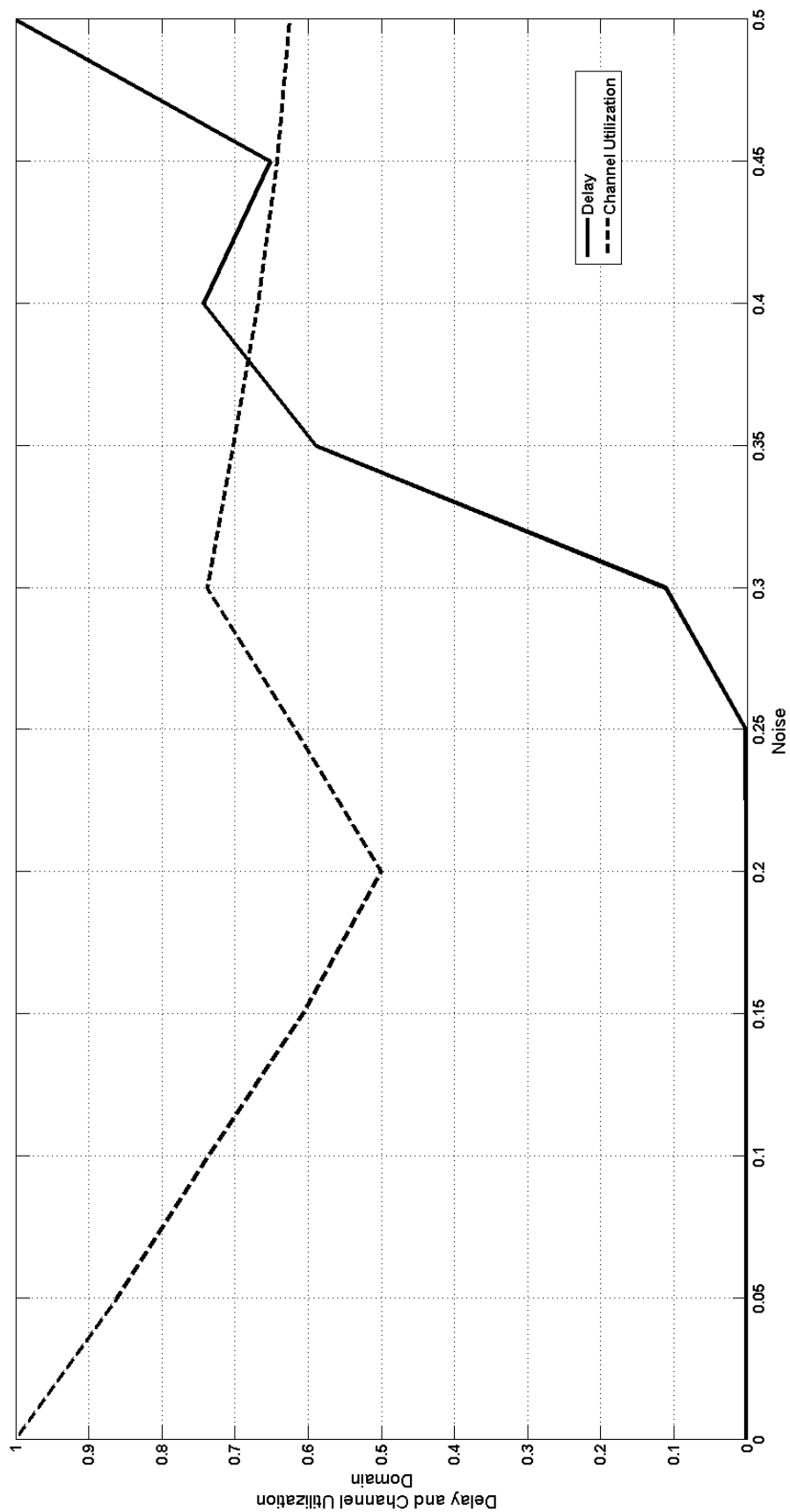


Figure 24. Noise versus domain of delay and channel utilization with one sensor using ENSURE 1.0

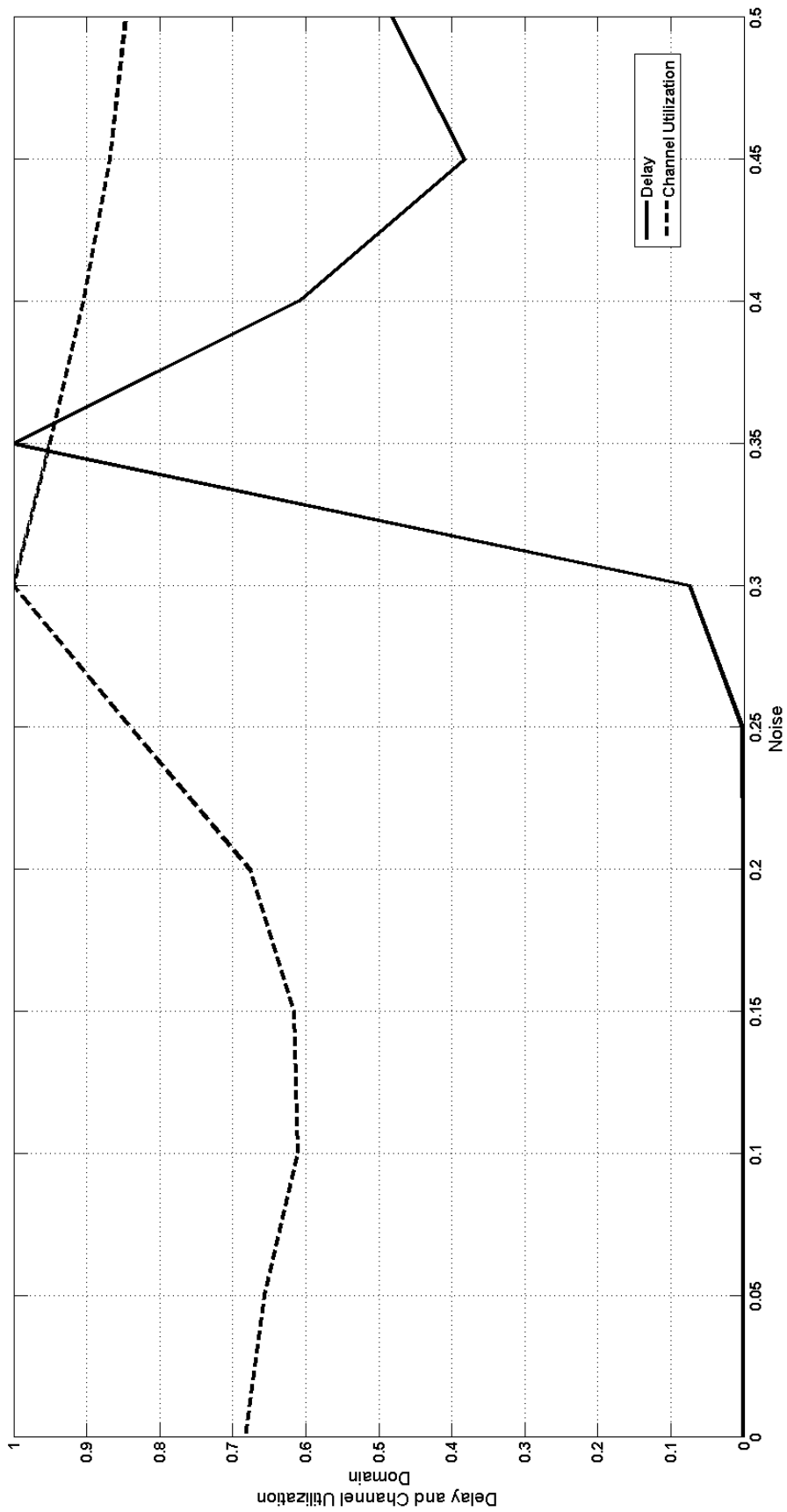


Figure 25. Noise versus domain of delay and channel utilization with two sensors using ENSURE 1.0

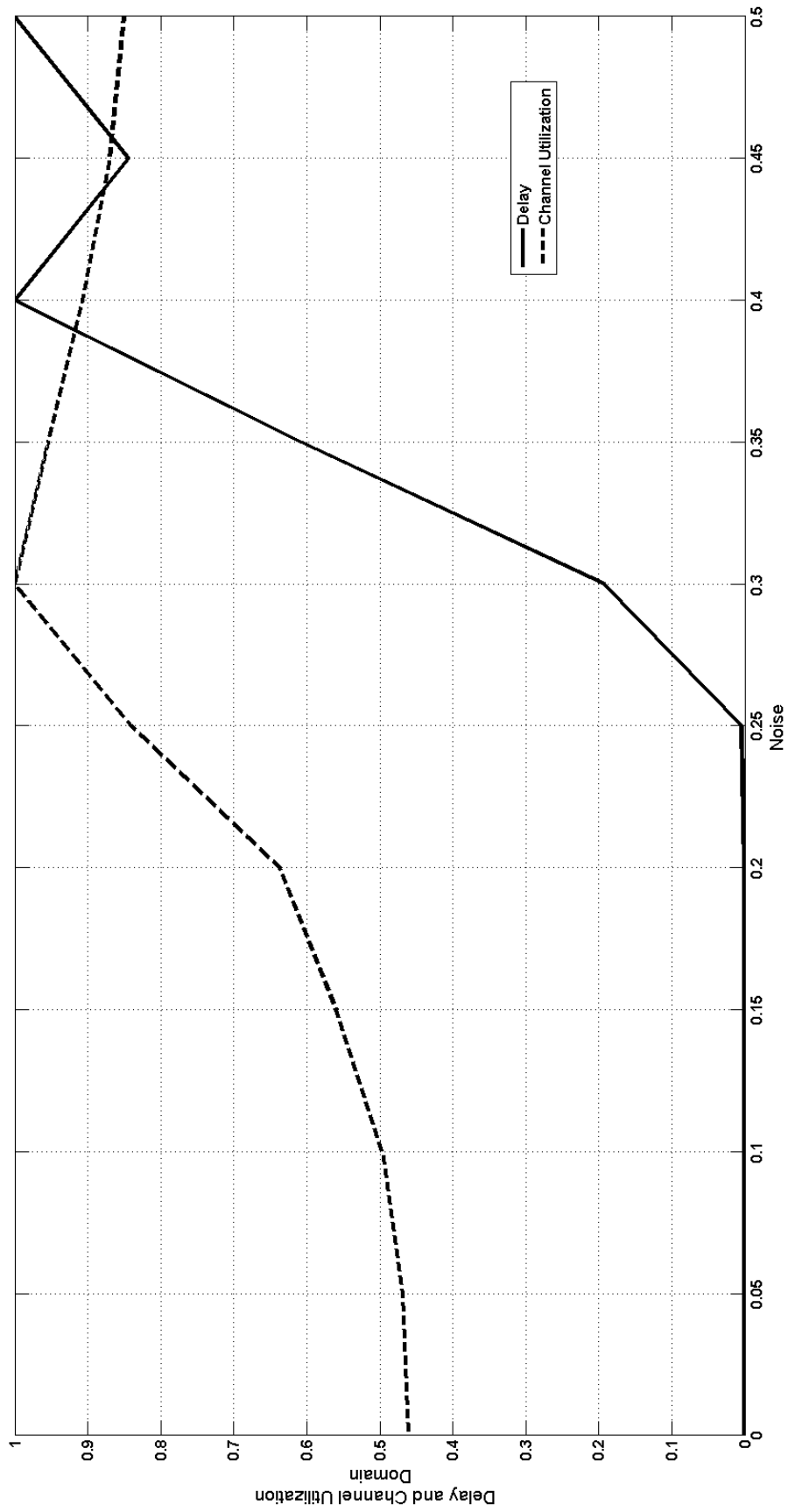


Figure 26. Noise versus domain of delay and channel utilization with three sensors using ENSURE 1.0

5.3 ENSURE 2.0: Channel Utilization

In ENSURE 1.0, industry's main concern of reliability was addressed; however, from a practical sense there must be a time restraint placed on the protocol. This is because the petroleum industry works in a real-time environment: if a reading on a temperature gauge comes back high, the base station can send signals to the valves to adjust the flow in an effort to cool it down or in a worst case situation, can completely shut of the pumps. Failure to make adjustments would have catastrophic results similar to what was seen in the 2010 British Petroleum explosion off the shore of Louisiana, which left workers dead.

ENSURE 2.0 improves on the base protocol by adding a limit on the number of times a reading can go through the retransmissions process. In the model used, we limited this number to an initial transmission and one round of retransmissions. Simulations were run again at noise levels of 0 to 50% in 5% steps. After each simulation the network conditions were recorded and then graphed for comparison purposes. In this section, the protocol will be examined based on the number of sensors; in later Chapters a comparison between protocols will be made.

5.3.1 Delay versus Channel Utilization

Since ENSURE 2.0 is limiting the number of times a packet can be cycled through the network, delay versus channel utilization measurement must be revisited. There are two possible effects that the new way of handling packets could have: 1) the channel utilization could increase because it is not creating more noise in the atmosphere or 2) the utilization could drop because packets are not given enough cycles to overcome the noise creating losses. In the second scenario one would expect to see an increase in channel utilization when adding reading sources.

In Figure 24, it can be observed that when the delay is over 0.25 seconds the channel utilization steadily approaches 10% when one sensor is online. The maximum delay is around 4.5 seconds over the 50,000 second simulation. In Figure 25, there is an addition of a sensor that leads to a maximum delay of 7 seconds but a minimum channel utilization of 20%. The pattern of the plot is the same as one sensor, just a smaller slope. When looking at three sensors, as in Figure 26, the pattern is repeated but the minimum channel usage is 30%. In conclusion, by adding a sensor a tradeoff can be made: an increase in delay for a gain in effective resource usage. All sensors are compared side by side in Figure 27 and shows that adding additional sensors does not hurt the channel usage capabilities.

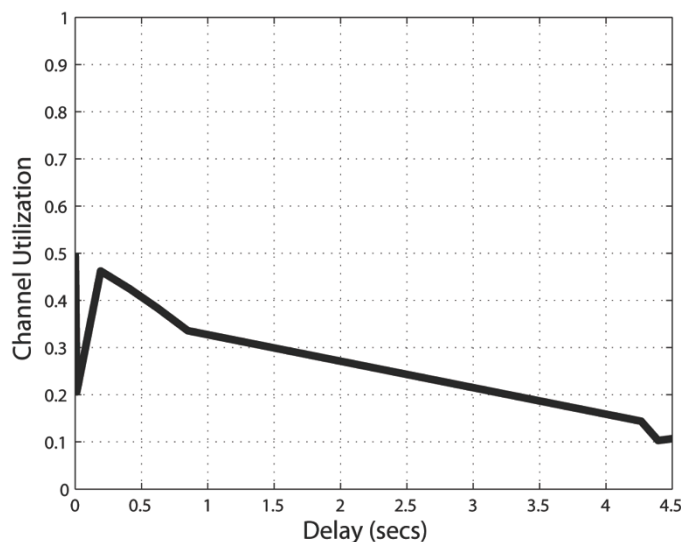


Figure 27. Delay versus channel utilization for one sensor using ENSURE 2.0

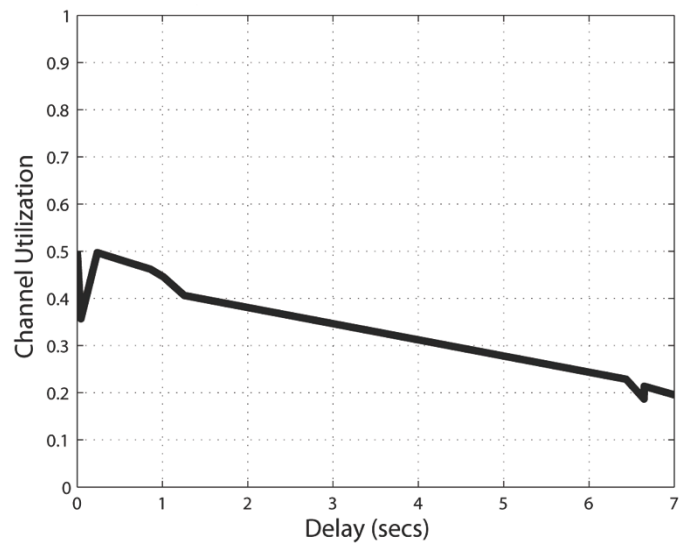


Figure 28. Delay versus channel utilization for two sensors using ENSURE 2.0

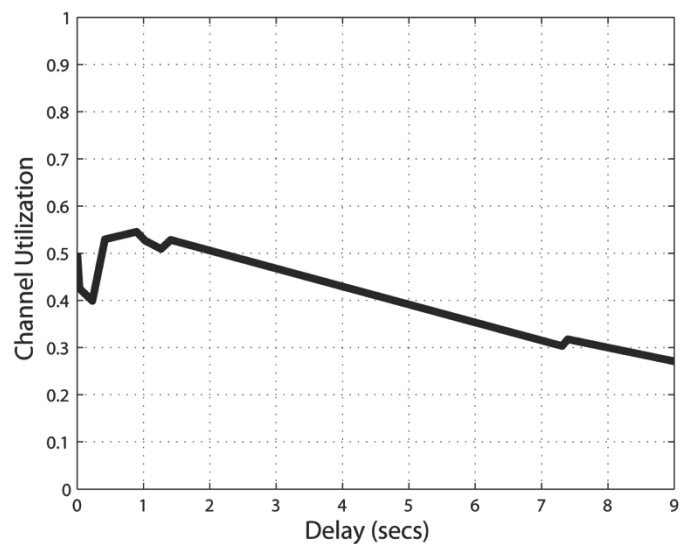


Figure 29. Delay versus channel utilization for three sensors using ENSURE 2.0

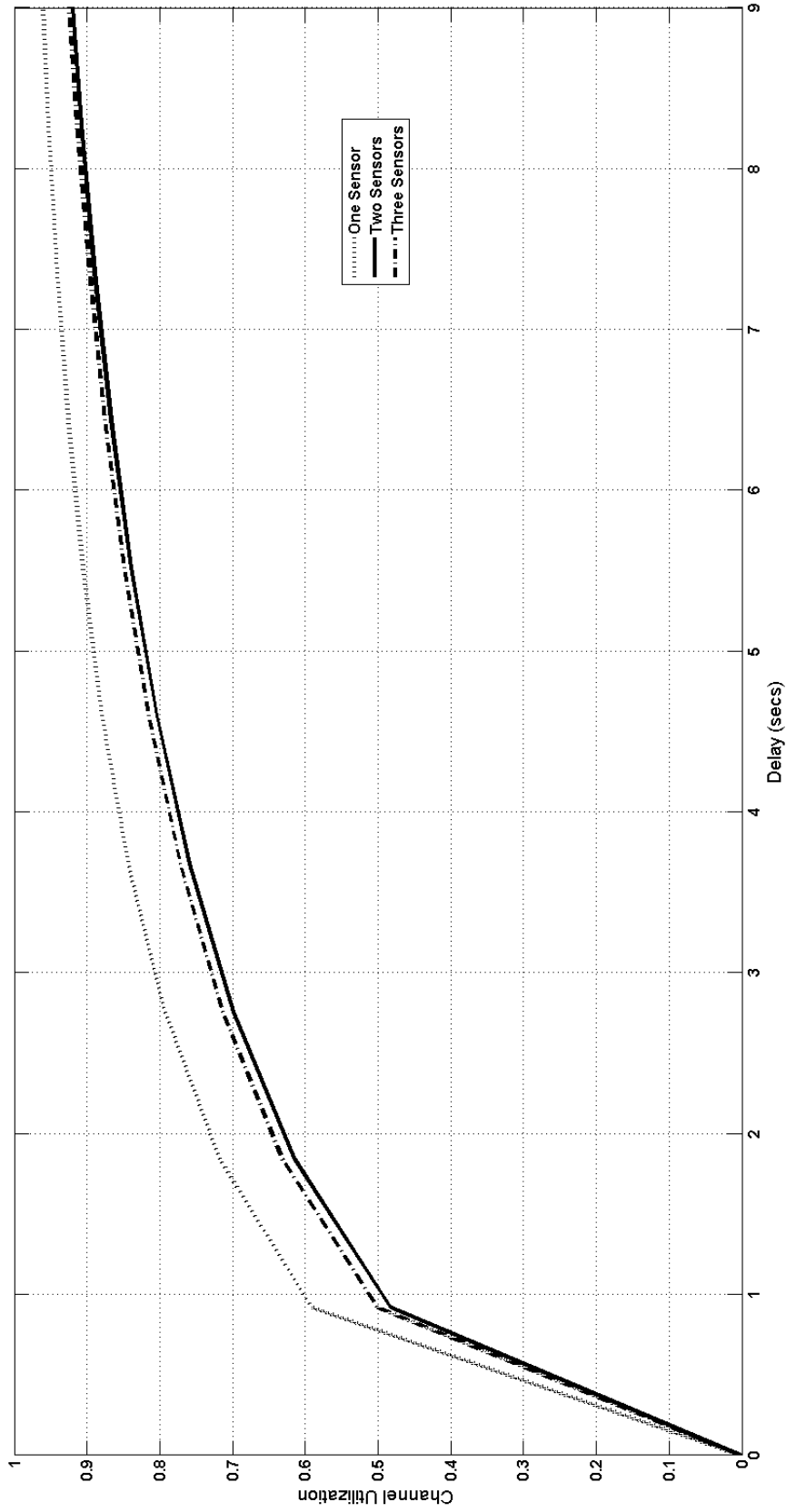


Figure 30. Comparison of delay versus channel utilization for all sensors using ENSURE 2.0

5.3.2 Noise versus Channel Utilization

In traditional protocols, channel noise and overall delay typically go hand in hand, but in ENSURE 2.0 this is not the case. Since it implements a floating window size which is dependent on the current reliability of the network, a larger noise level does not automatically produce lower channel utilization. Figure 28, which represents one sensor, shows that in a noiseless network 50% channel usage is achieved. The reason this is the maximum channel utilization is because the protocol assumes a 50% PLR from the beginning and then adjusts down. With this said, the first transmission sends packets twice in attempt to overcome any loss in the initial transmission. With one sensor, as seen here, when noise levels are under 25%, each added noise affects it much more than noise added to a channel over 25% noise. With two sensors, Figure 29, noise levels at and below 30% keep a fairly consistent usage between 40 and 50%. Between 30 and 35% each added noise reduces the resource usage amount dramatically, while levels above 35% produce slight changes hovering around 20% usage. The simulation process then continues to three sensors, Figure 30, where interesting results occur. With lower noise levels, below 25%, the 50% maximum utilization is no longer in place; the protocol initially overcomes this barrier, then hovers around it. However, when the 25% noise level is exceeded, there is a rapid decrease in meaningful transmissions. Once again, a sensor performance comparison, Figure 21, shows that more sensors equal more performance.

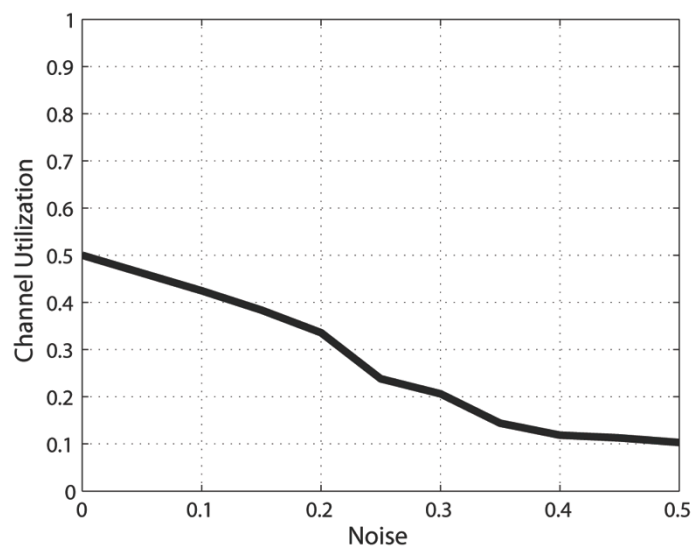


Figure 31. Noise versus channel utilization for one sensor using ENSURE 2.0

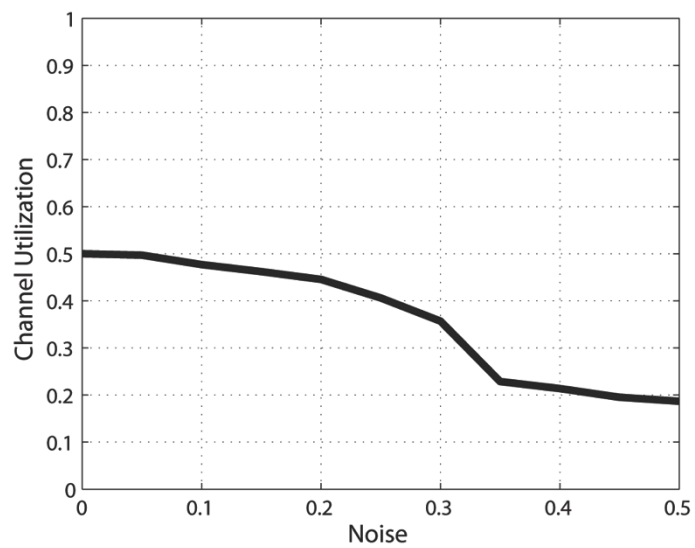


Figure 32. Noise versus channel utilization for two sensors using ENSURE 2.0

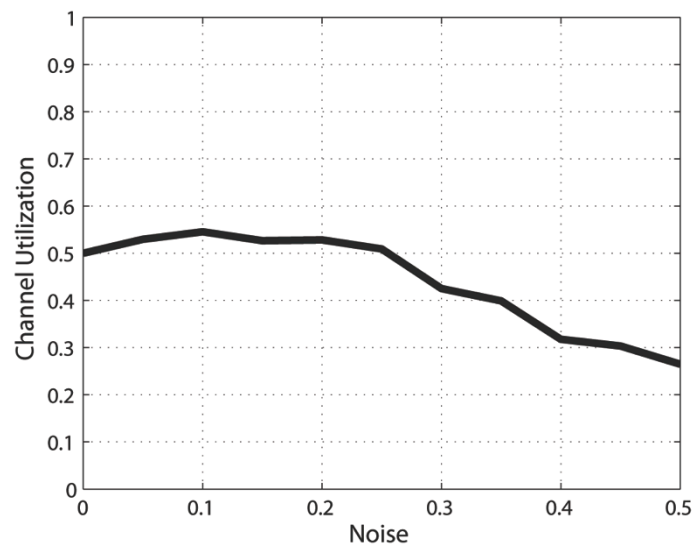


Figure 33. Noise versus channel utilization for three sensors using ENSURE 2.0

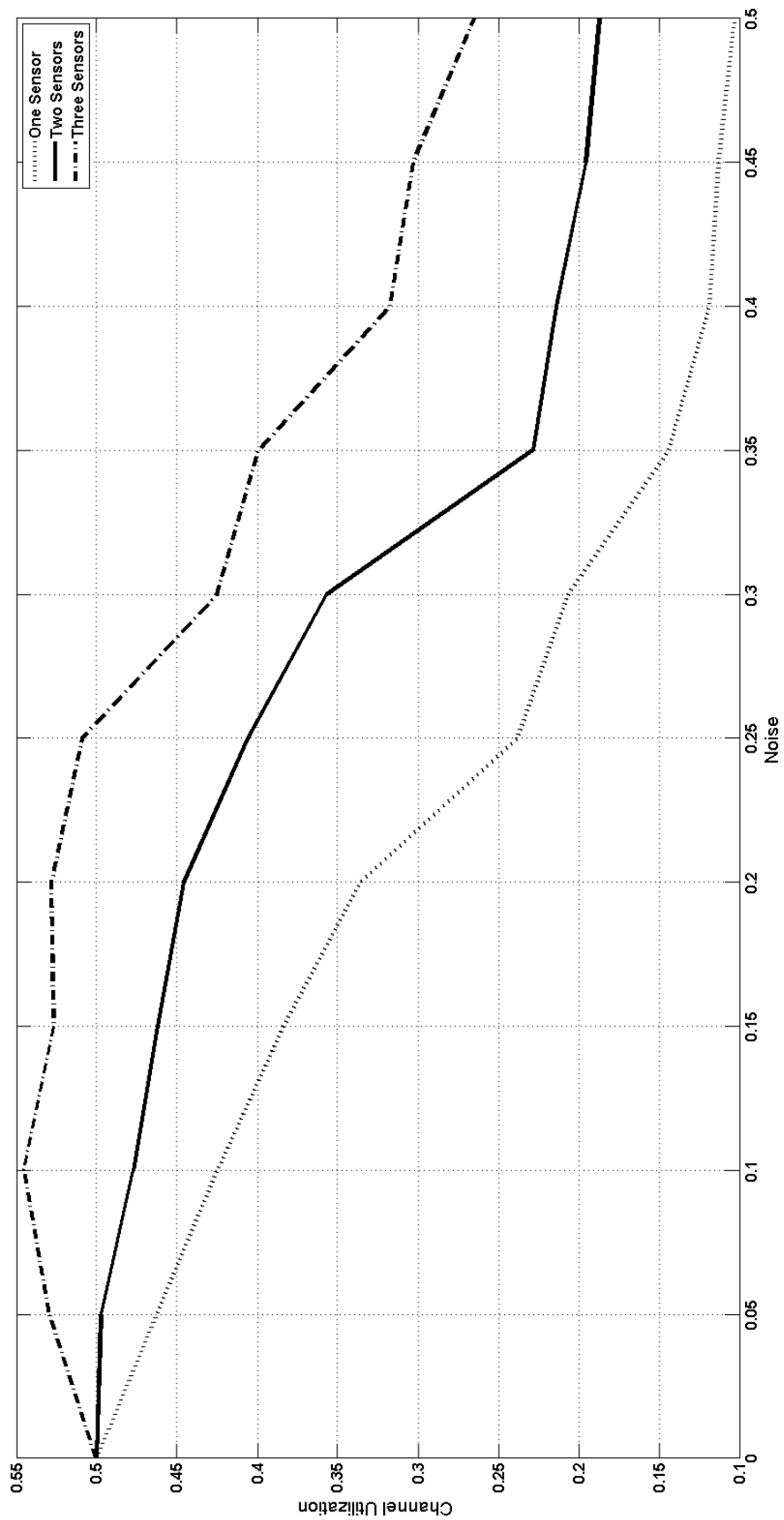


Figure 34. Comparing noise versus channel utilization for all sensors using ENSURE 2.0

5.3.3 Channel Utilization versus Reliability

When examining the way ENSURE 2.0 reacts to a lost transmission, the ability of the protocol to respond in a way that minimizes a total loss of the reading must be considered. One way this can be verified is to examine a graph of the reliability versus channel utilization; theoretically the percentage of channel usage will decrease at a linear rate when the reliability decreases. Note that complete reliability is represented by 0% on the graphs. Also, the more sensors added the less the slope of the graph changes. With one sensor, Figure 32 demonstrates that the total change in channel utilization to be roughly 40%. Thus it is extremely responsive to a lost transmission. Adding an additional sensor, as in Figure 33, shows that the network is still responsive but with a 30% range of change. With these first two simulations the resource usage is always diminishing, however, adding one more sensor changes this pattern. Figure 34 graphs three sensors and shows that with low noise levels, i.e. less than 15%, the channel is taken advantage of more than the initial 50%. After the threshold is reached the patterns of the previous simulations are present with a range of 20% change in usage. A complete comparison on this measure, Figure 35, verifies that three sensors make the most of the given channel.

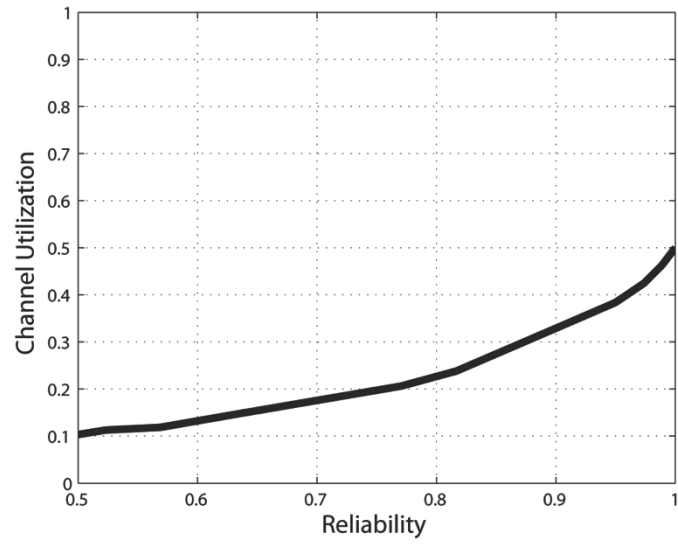


Figure 35. Reliability versus channel utilization for one sensor using ENSURE 2.0

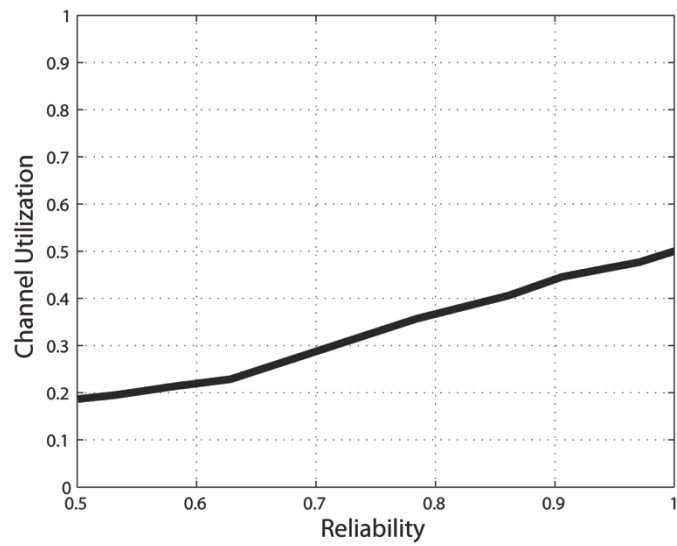


Figure 36. Reliability versus channel utilization for two sensors using ENSURE 2.0

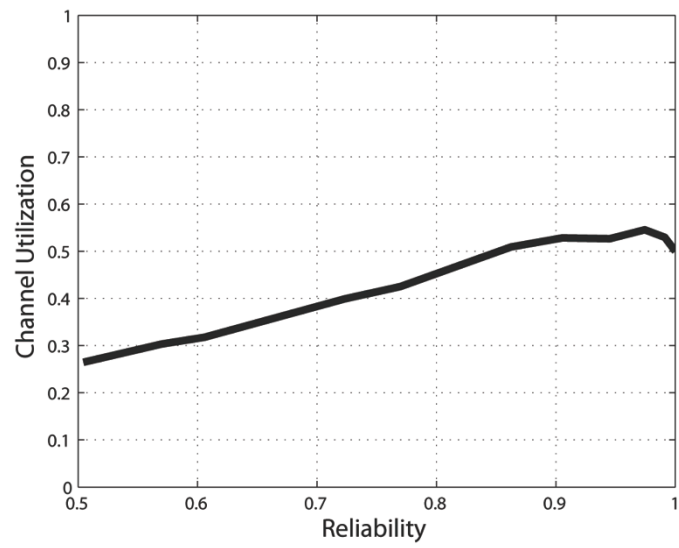


Figure 37. Reliability versus channel utilization for three sensors using ENSURE 2.0

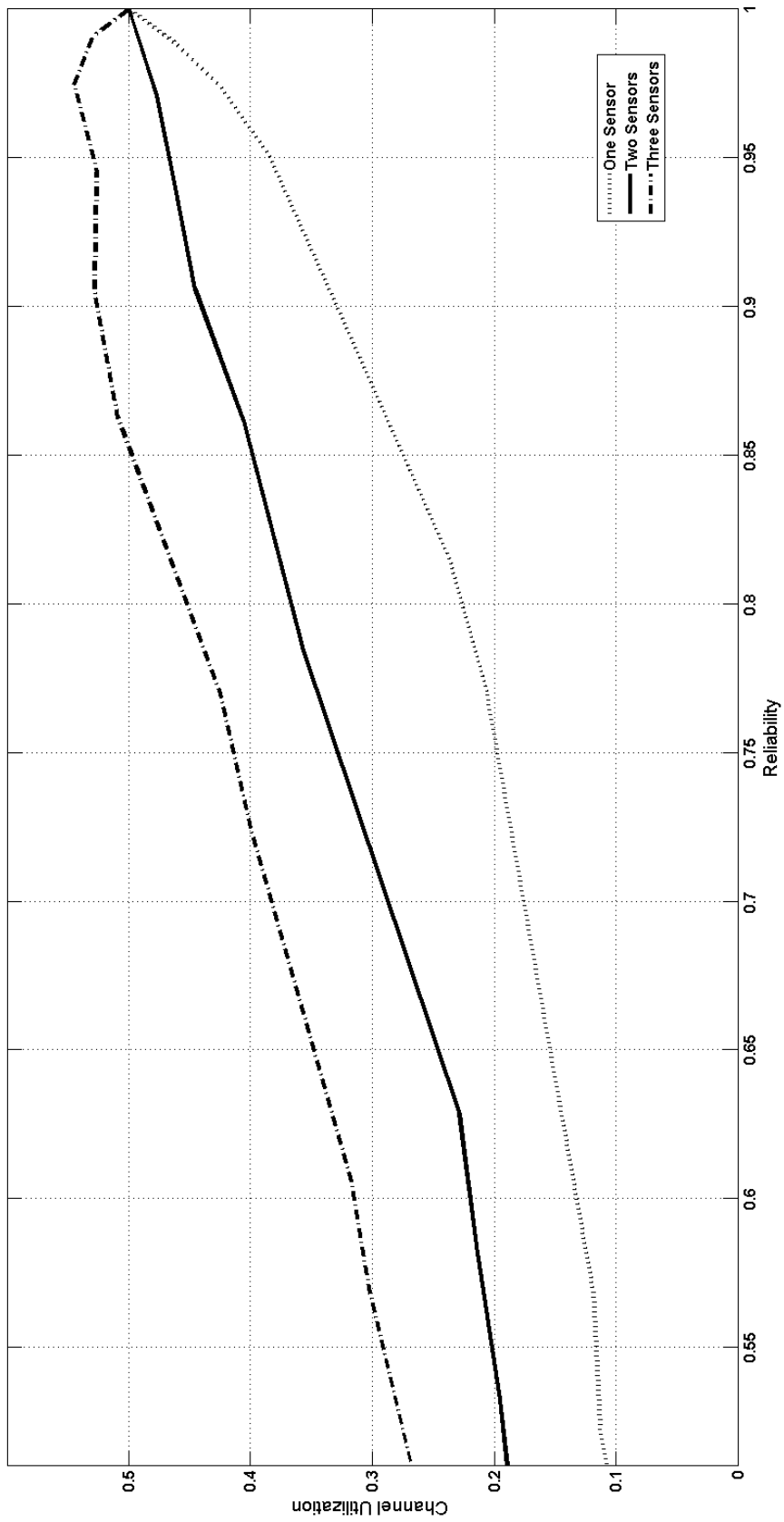


Figure 38. Comparison of reliability versus channel utilization for all sensors using ENSURE 2.0

5.3.4 Noise versus Retransmissions

Given the previous measures, one must now verify that the protocol is making a reasonable attempt to overcome environmental noise. With ENSURE 1.0, it was proven that there were no lost packets, but without the added time restraint it could have retransmitted the same reading many times before a clean packet was received. In ENSURE 2.0, analysis of this measure will provide a validation of efficiency. When looking at a network with one reading source, as in Figure 36, a clear drop in the amount of retransmissions occurs at 35% noise levels. This shows that the network is effectively adjusting when a lost transmission is detected; it is not automatically sending the most retransmissions but determining what the ideal amount is. However, this also indicates that at the peak of 30% noise the protocol is retransmitting more than necessary. While this is also true in the two-sensor network (Figure 37), at 30% noise it is not as drastic. This indicates the ability to adjust with a minimum of overcompensation. Figure 38 shows that with three sensors the protocol reverts back to the same situation as with one sensor, it is sending too many packets at 35% to ensure no lost readings. When graphing all the sensors in one graph, as shown in Figure 39, there is not an optimal setup for all situations, if the focus was keeping the retransmissions to a minimum.

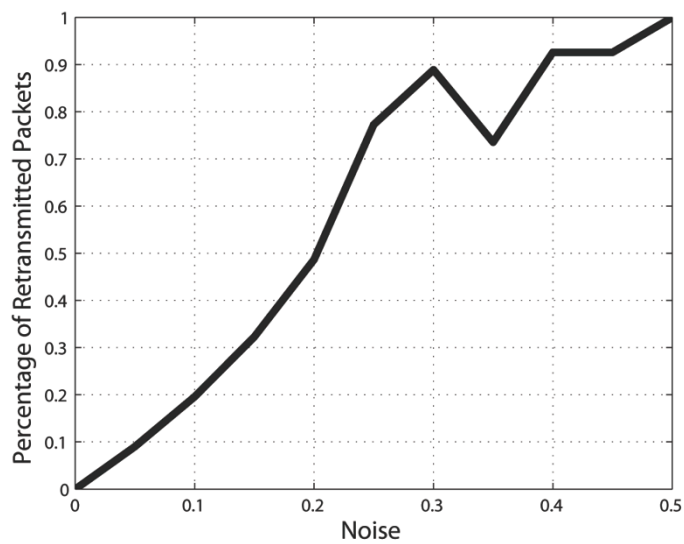


Figure 39. Noise versus number of retransmissions required for one sensor using ENSURE 2.0

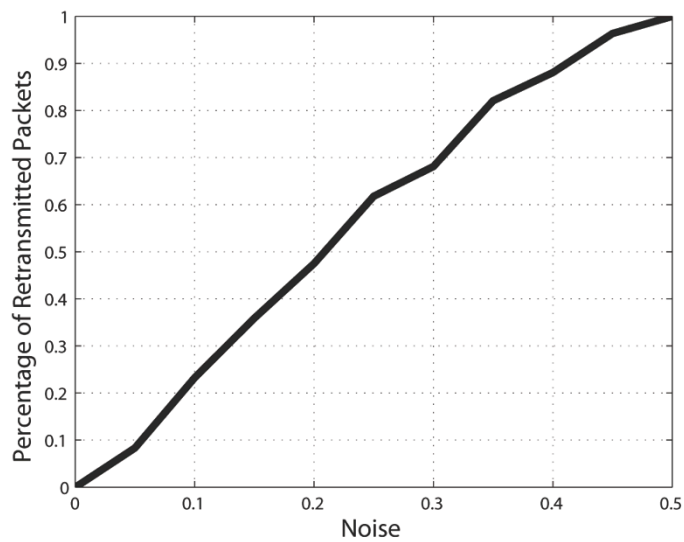


Figure 40. Noise versus number of retransmissions required for two sensors using ENSURE 2.0

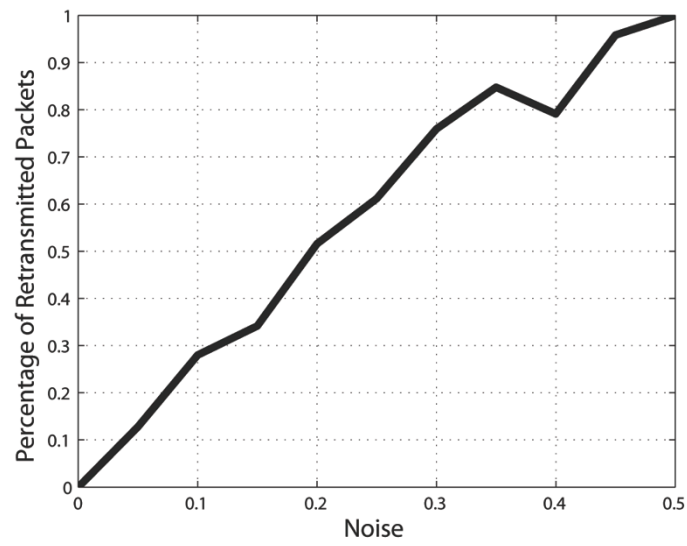


Figure 41. Noise versus number of retransmissions required for three sensors using ENSURE 2.0

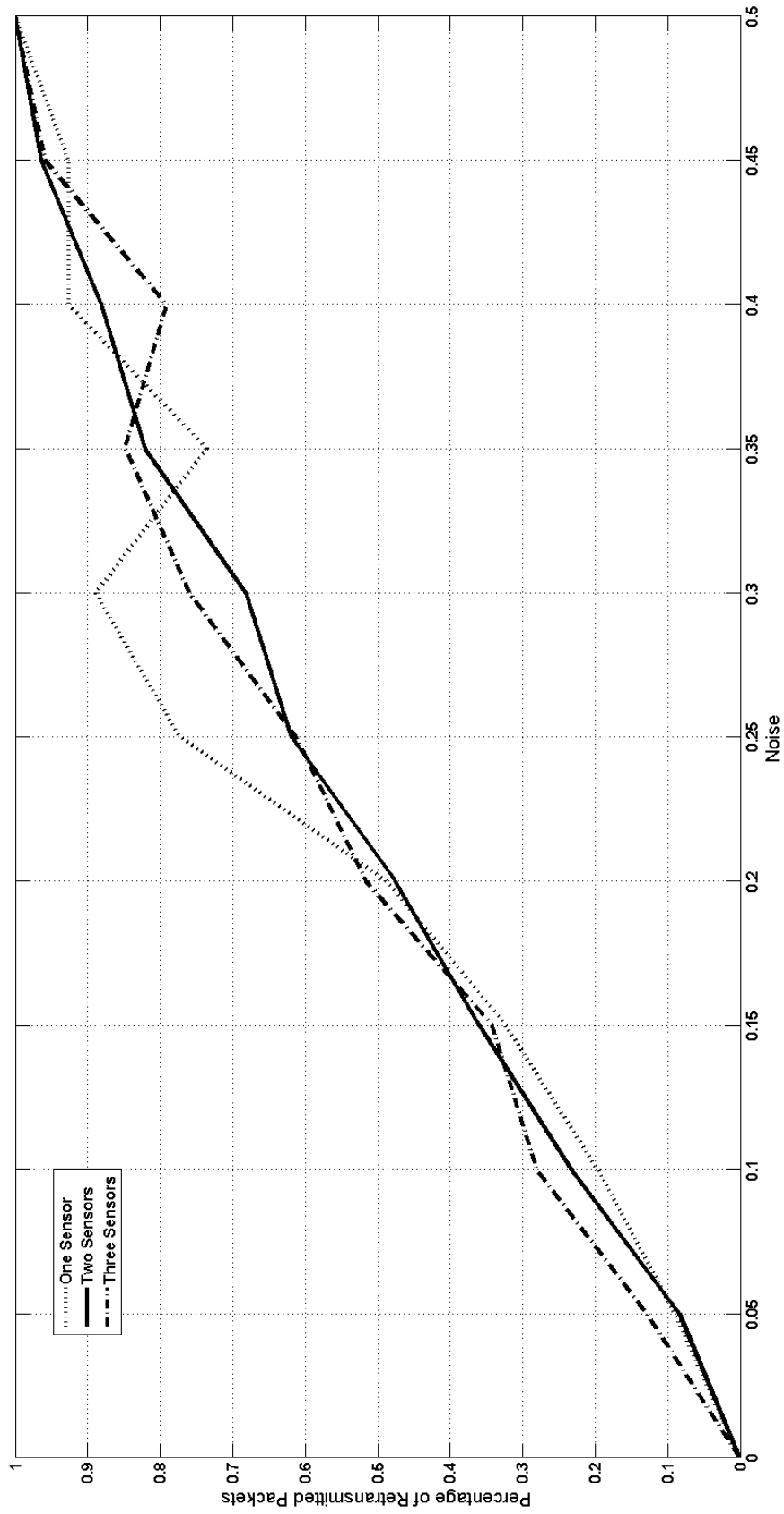


Figure 42. Comparison of noise versus number of retransmissions required for all sensors using ENSURE 2.0.

5.3.5 Reliability versus Retransmissions

ENSURE 2.0 is designed to produce high reliability with a minimum delay. One way to make sure the protocol is working as planned is to graph the reliability of a channel against the total number of retransmissions the simulation sent. One would expect the lower the reliability, the lower the number of transmissions sent. As with the previous measure, Figure 40 shows that there is a sharp drop in the number of retransmissions at 25% noise on a one sensor network. The results mirror those seen in the measurement of noise versus retransmissions and indicate the protocol is over-compensating for losses at a reliability level of 25%. With two sensors, Figure 41, a smoother response to a loss is shown, meaning there is no point that the protocol sends too many retransmissions per loss. When running three sensors (Figure 42), however, there are points that too many retransmissions are sent at both 5% and 25%. As when looking at noise versus retransmissions, there is no clear answer as to the best setup (Figure 43).

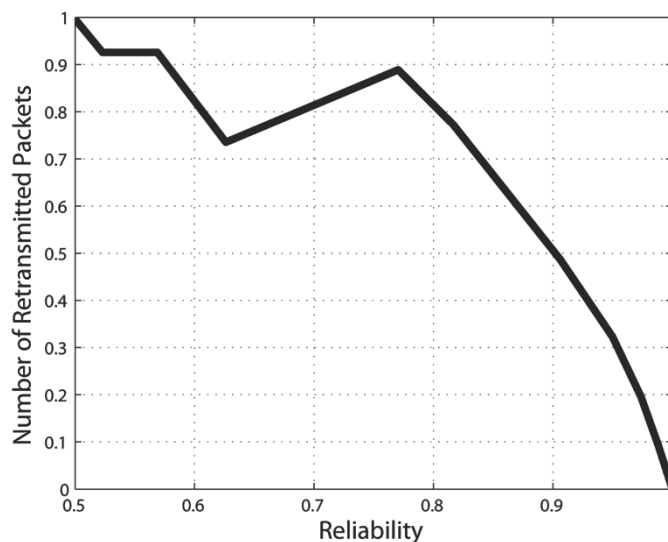


Figure 43. Reliability versus number of retransmissions required for one sensor using ENSURE 2.0

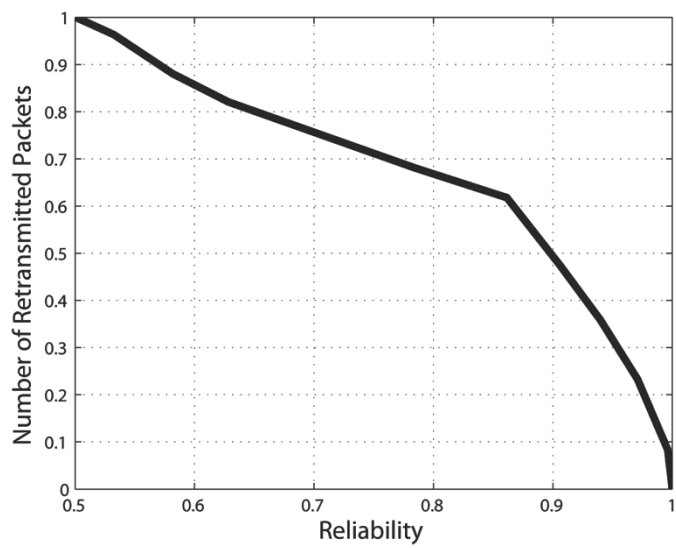


Figure 44. Reliability versus number of retransmissions required for two sensors using ENSURE 2.0

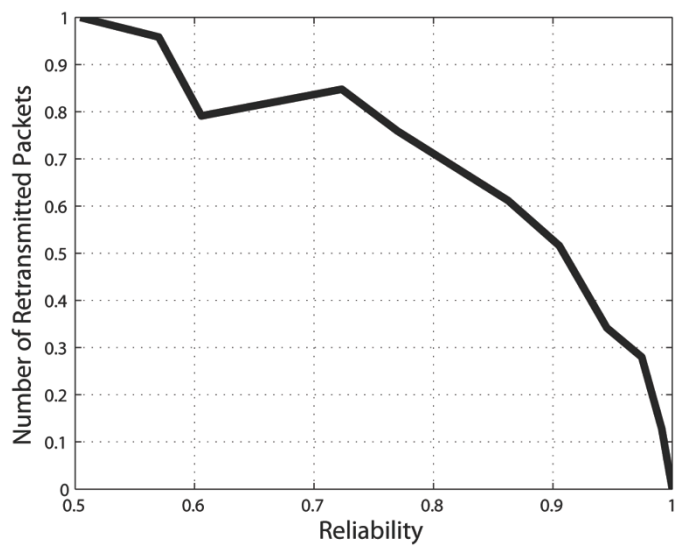


Figure 45. Noise versus number of retransmissions required for three sensors using ENSURE 2.0

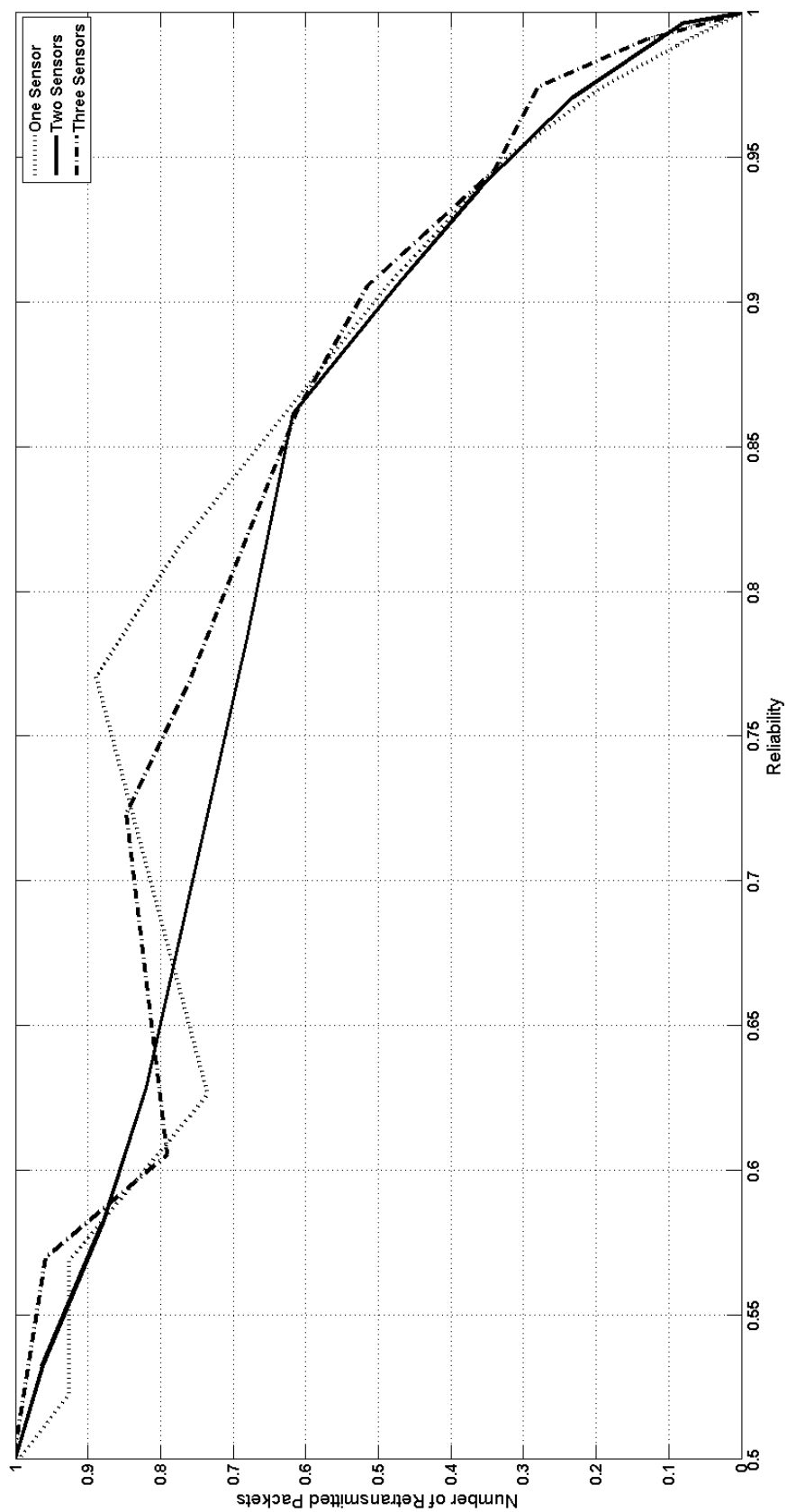


Figure 46. Comparison of noise versus number of retransmissions required for all sensors using ENSURE 2.0

5.3.6 Noise versus Packets Loss

The main goal of ENSURE is to minimize loss and the final two measures examine these all important numbers. Since the channel interference is the cause of a packet loss, examining the noise level against the packet loss is an ideal starting point. The results of a one sensor simulation prove there are no complete losses on the network, i.e. all readings were recoverable by the end of the simulated time. The graphs for a two and three sensor network, as shown in Figures 44 and 45 respectively, reflect the same issue: the network has a point where it does not adjust well enough to prevent losses. This could be caused by not switching to a higher retransmission rate soon enough or by a toggling of rates that does not consistently cover a packet loss. In most cases, extreme changes in noise only take place in the petrochemical environment when the equipment setup is changed. This lack of change makes the comparing all simulation results for noise versus packet loss extremely important. As seen in Figure 46, adding sensors into a network running ENSURE 2.0 dramatically increases the number of packets lost around the network.

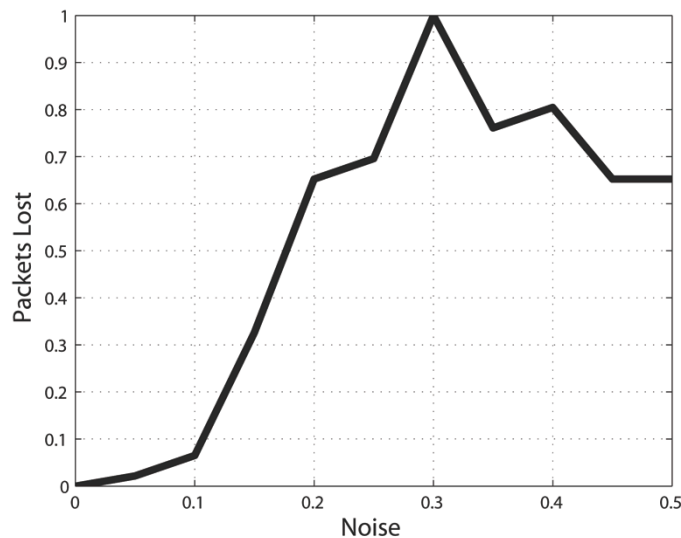


Figure 47. Noise versus number of packets lost for two sensors using ENSURE 2.0

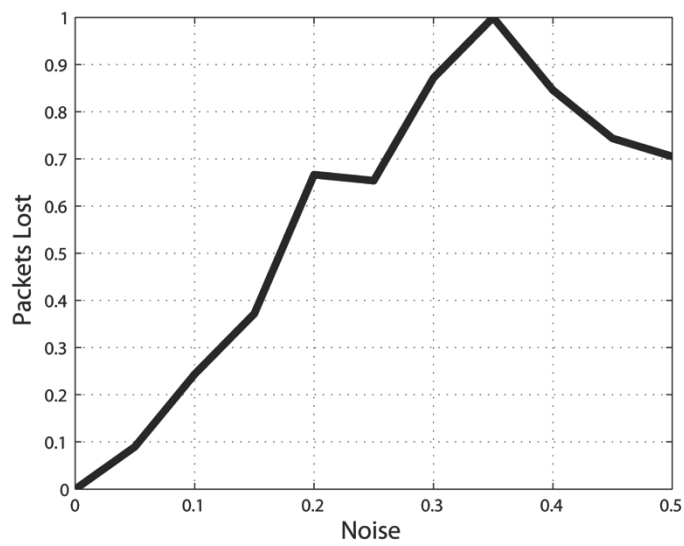


Figure 48. Noise versus number of packets lost for three sensors using ENSURE 2.0

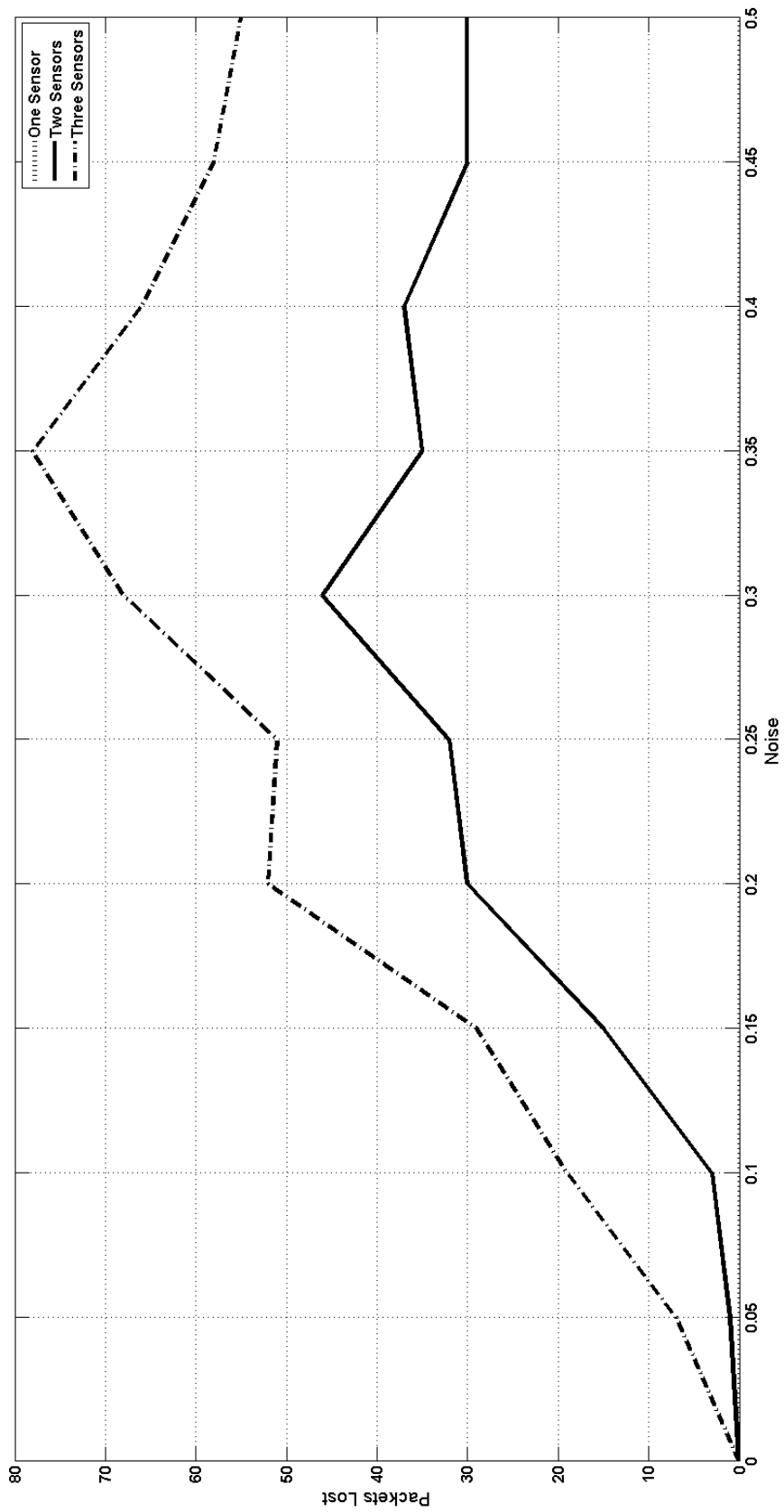


Figure 49. Comparison of noise versus number of packets lost for all sensors using ENSURE 2.0

5.3.7 Reliability versus Packet Loss

Finally, one must consider reliability versus the number of packets lost. At this point it is important to note the subtle difference in noise and reliability and it can be summed up in a simple sentence: Noise is a constant that controls the probability of losing a packet, while reliability is a dynamic measure of the number of packets lost over the lifetime of the simulation. With this said, it is interesting to note that the graphs for reliability reflect those of noise when plotted against loss. The main difference is where the losses peak with when two or more sensors are used as one sensor maintains complete reliability. To further analyze this, consider the equations used to design ENSURE 2.0. A network with two reading sources, Figure 47, has an initial window of 4; at each loss the protocol adjusts the number of retransmissions it sends. With that in mind, the reliability cut-off points are 25%, 50% and 75% of the maximum. Earlier documentation establishes the maximum reliability as 50% because the initial transmission is two packets for each reading obtained. Thus, one should see clear reduction in losses following 0.125, 0.25 and 0.375 reliability on the graph, which we do. In Figure 48, which graphs three sensors, the cut off points should be 0.84, 0.17, 0.25, 0.34 and 0.42, and again reductions are present at these points. Note that the numbers represented are normalized to a maximum loss of 46 packets on a two-sensor setup and 78 with three sensors. Based on these results, the protocol is clearly working to accomplish the goal and losses are at a minimum. Plotting all the simulations in one graph, Figure 49, reinforces this point.

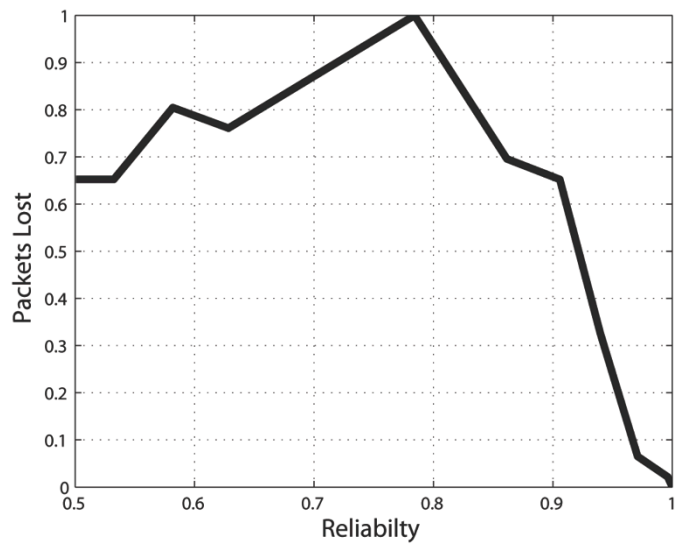


Figure 50. Reliability versus number of packets lost for two sensors using ENSURE 2.0

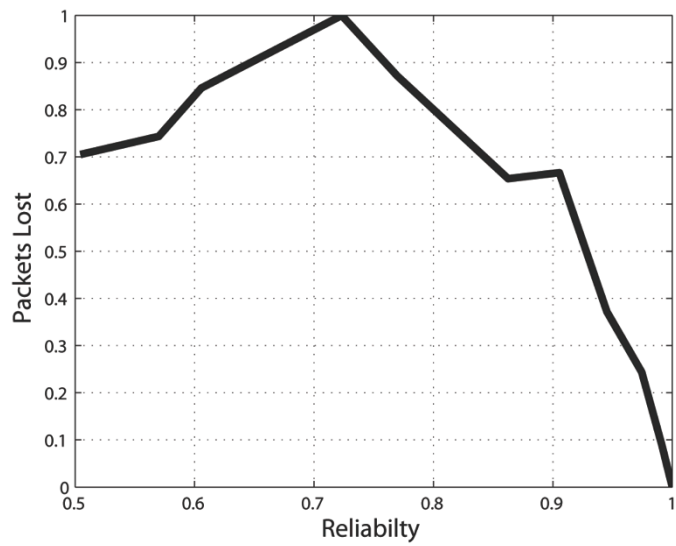


Figure 51. Reliability versus number of packets lost for three sensors using ENSURE 2.0

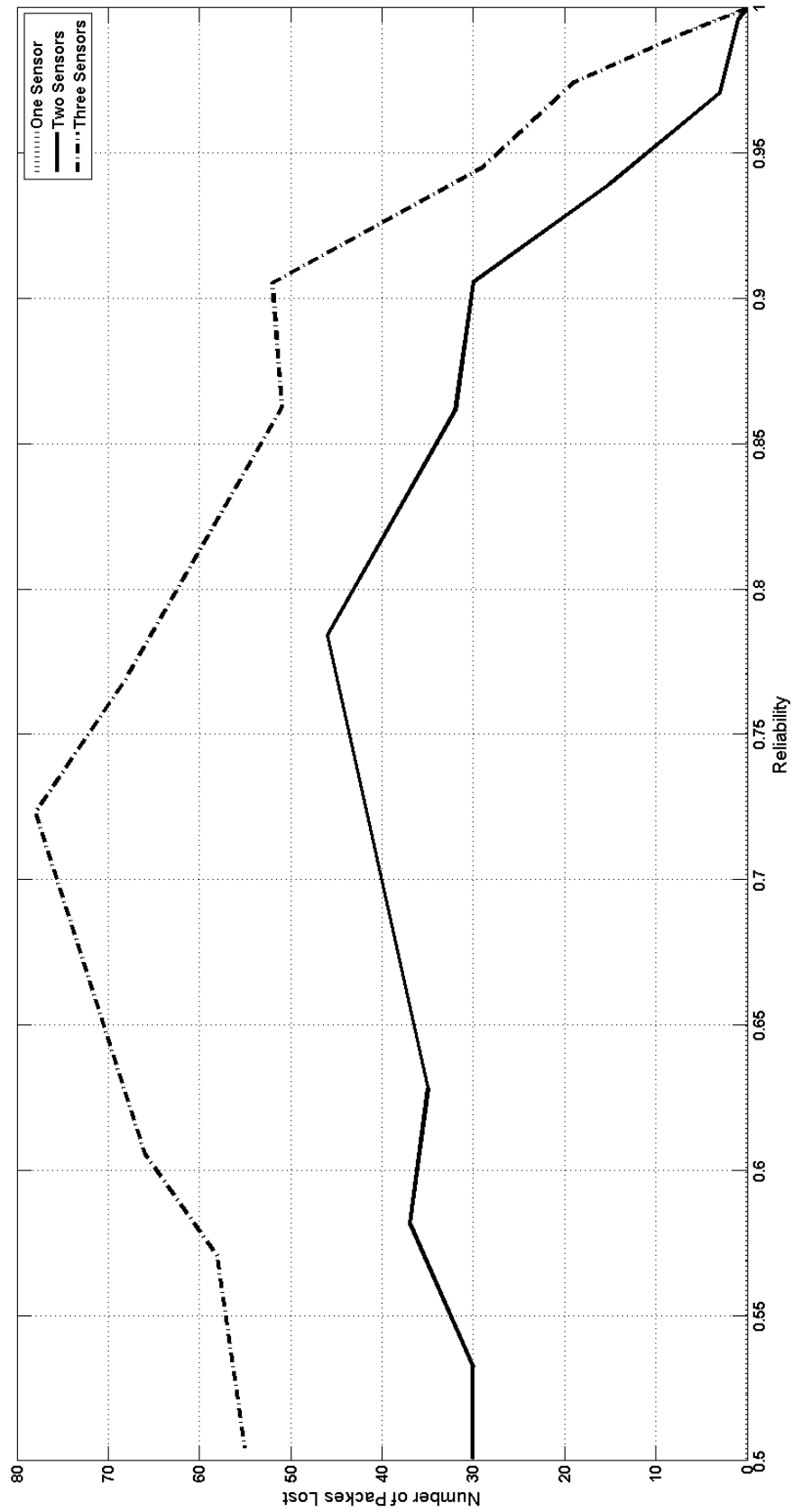


Figure 52. Comparison of reliability versus number of packets lost for all sensors using ENSURE 2.0

5.3.8 Noise versus Domain of Delay and Channel Utilization

Finally, combining two measures, delay and channel utilization, into the same domain and graphing them against the noise level allows one to recognize patterns. These patterns can be used to determine acceptable parameters when considering setting up a wireless network in a plant. Unlike ENSURE 1.0, 2.0 plots with low delays and high throughput are good. When the two measures begin to form natural clusters, this indicates that the setup is less than ideal. With one sensor, Figure 50, a cluster forms to the right of the measurements crossing between 30 and 35% noise. This indicates that once the delay outweighs the channel utilization, other factors, such as loss amounts, must be heavily considered to determine the worth of the setup. The same can be said of the two sensor graph (Figure 51), though by adding a larger window size the measurements cross later, at 40% noise. With three sources of readings (Figure 52), the criteria change a bit. Since the two measurements never cross, a simple distance measurement can be used. The lower the calculated difference the more actual packet loss will play into decisions.

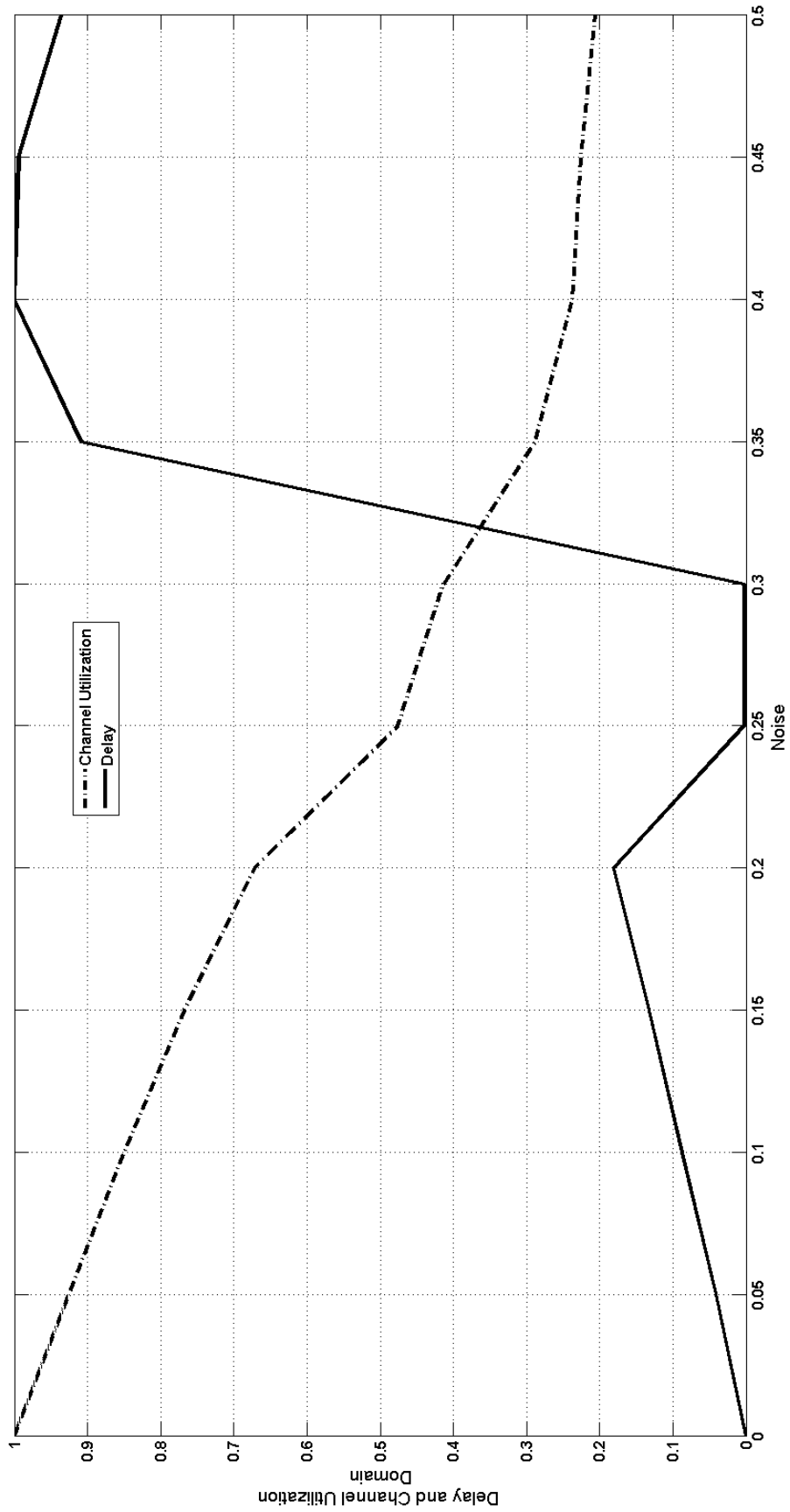


Figure 53. Noise versus domain of delay and channel utilization for one sensor using ENSURE 2.0

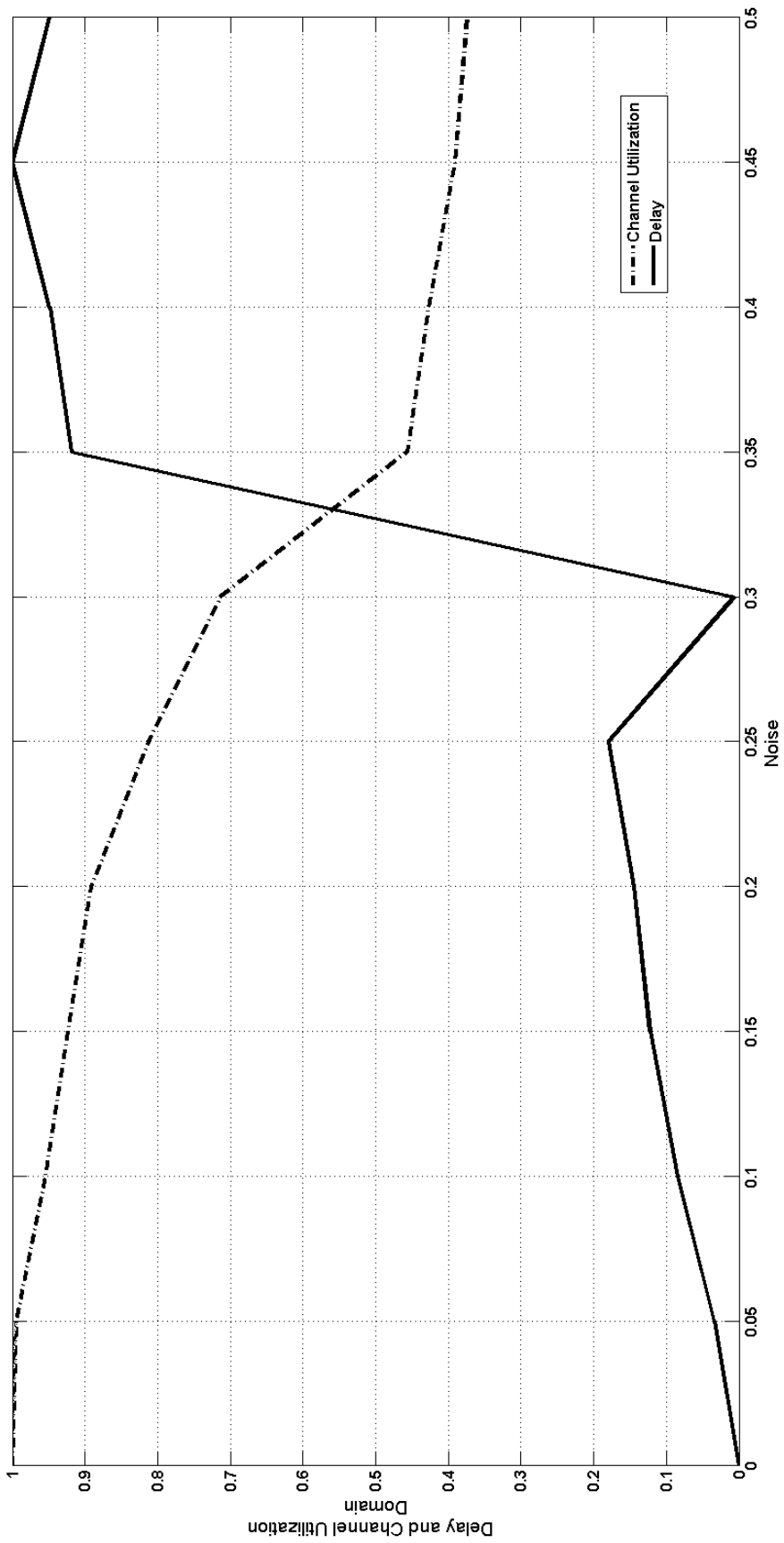


Figure 54. Noise versus domain of delay and channel utilization for two sensors using ENSURE 2.0

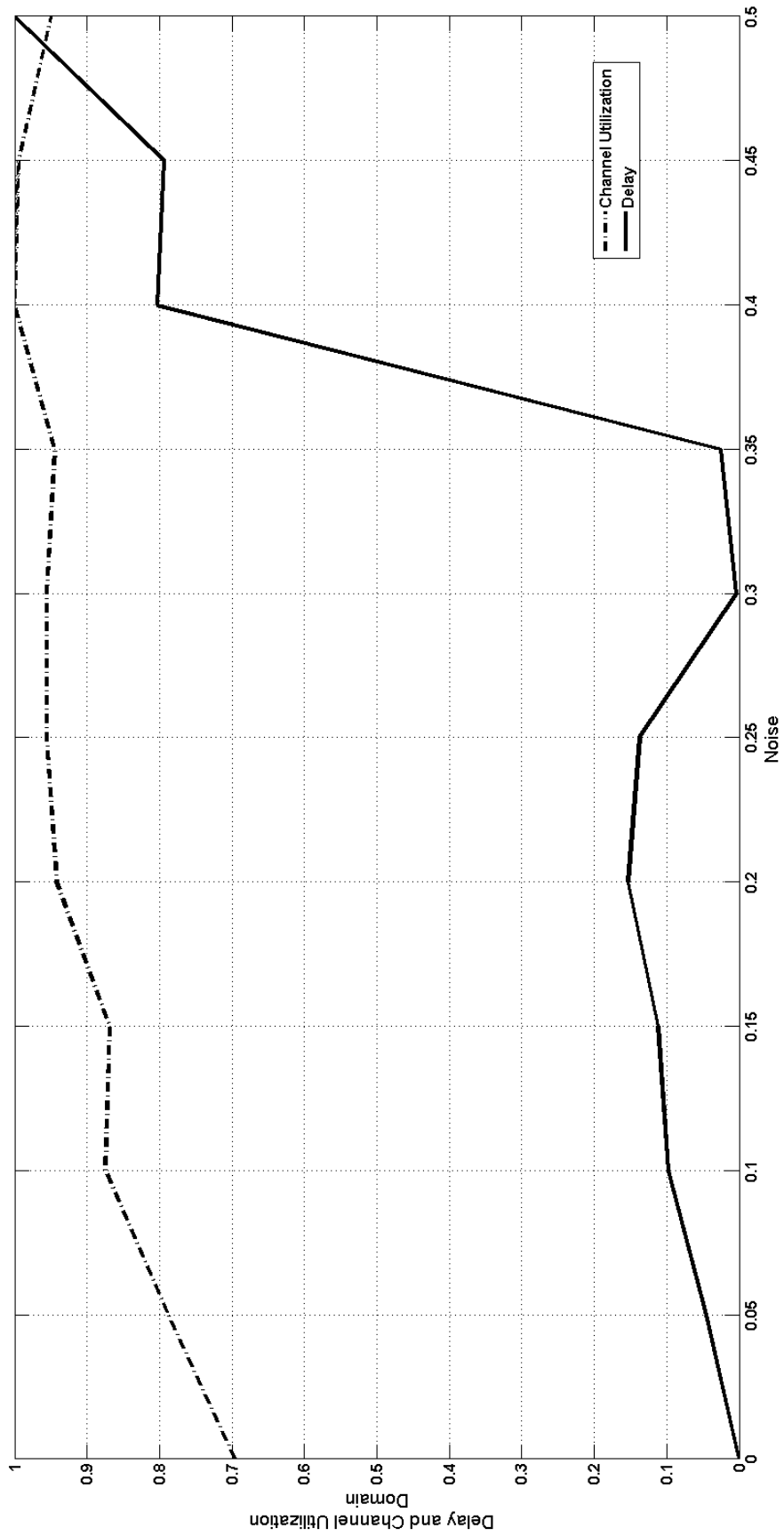


Figure 55. Noise versus domain of delay and channel utilization for three sensors using ENSURE 2.0

5.3.9 Reliability versus Domain of Delay and Channel Utilization

The properties of ENSURE 2.0 open the protocol to the possibility of a loss. With the added variable one must look at the reliability of the domain as well. In Figures 53 - 55, a clear separation in points is seen. The points to the left indicate that the network engineer must carefully consider packet loss versus delay when determining if this is a fair protocol for his situation.

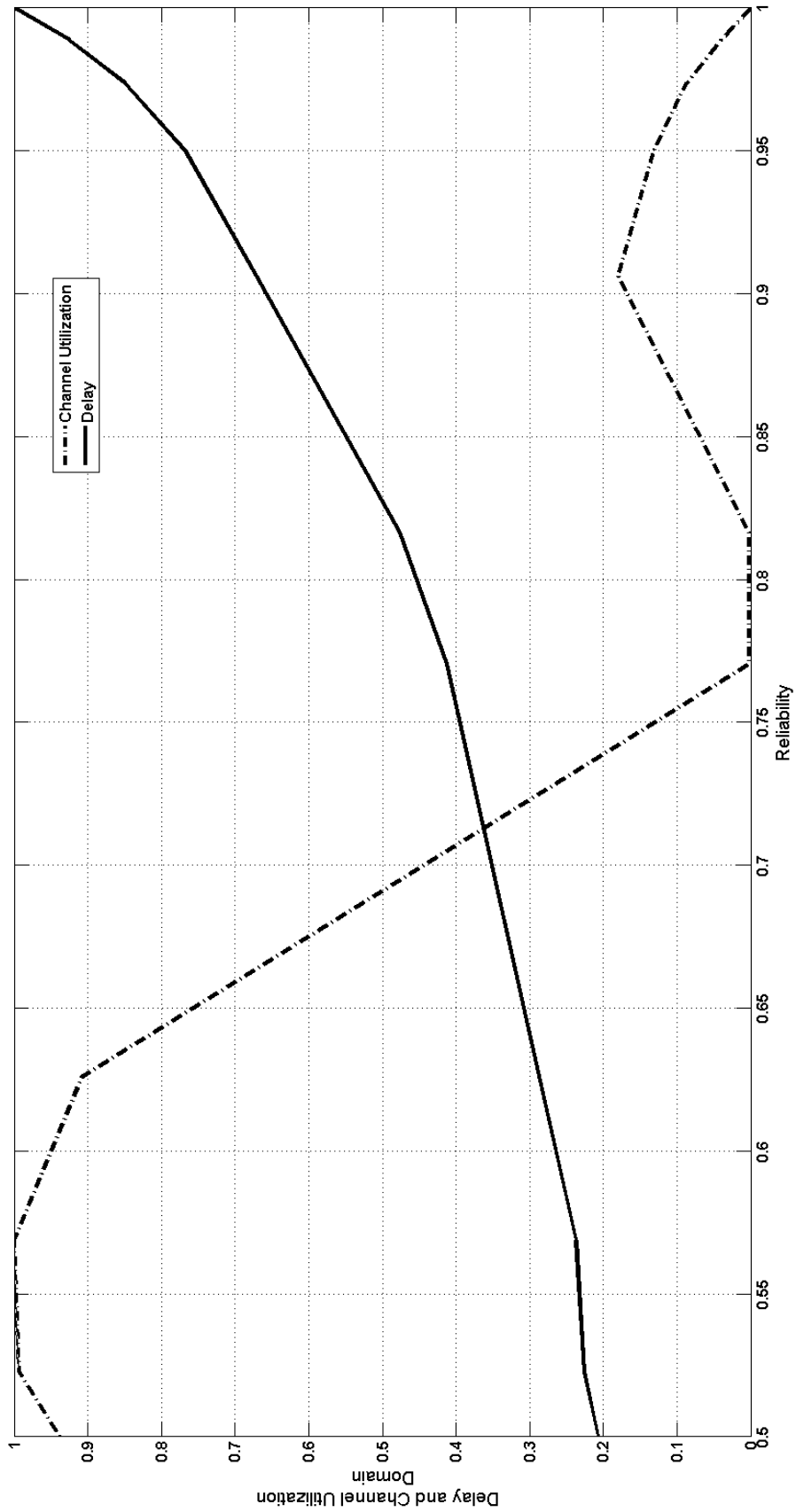


Figure 56. Reliability versus domain of delay and channel utilization for one sensor using ENSURE 2.0

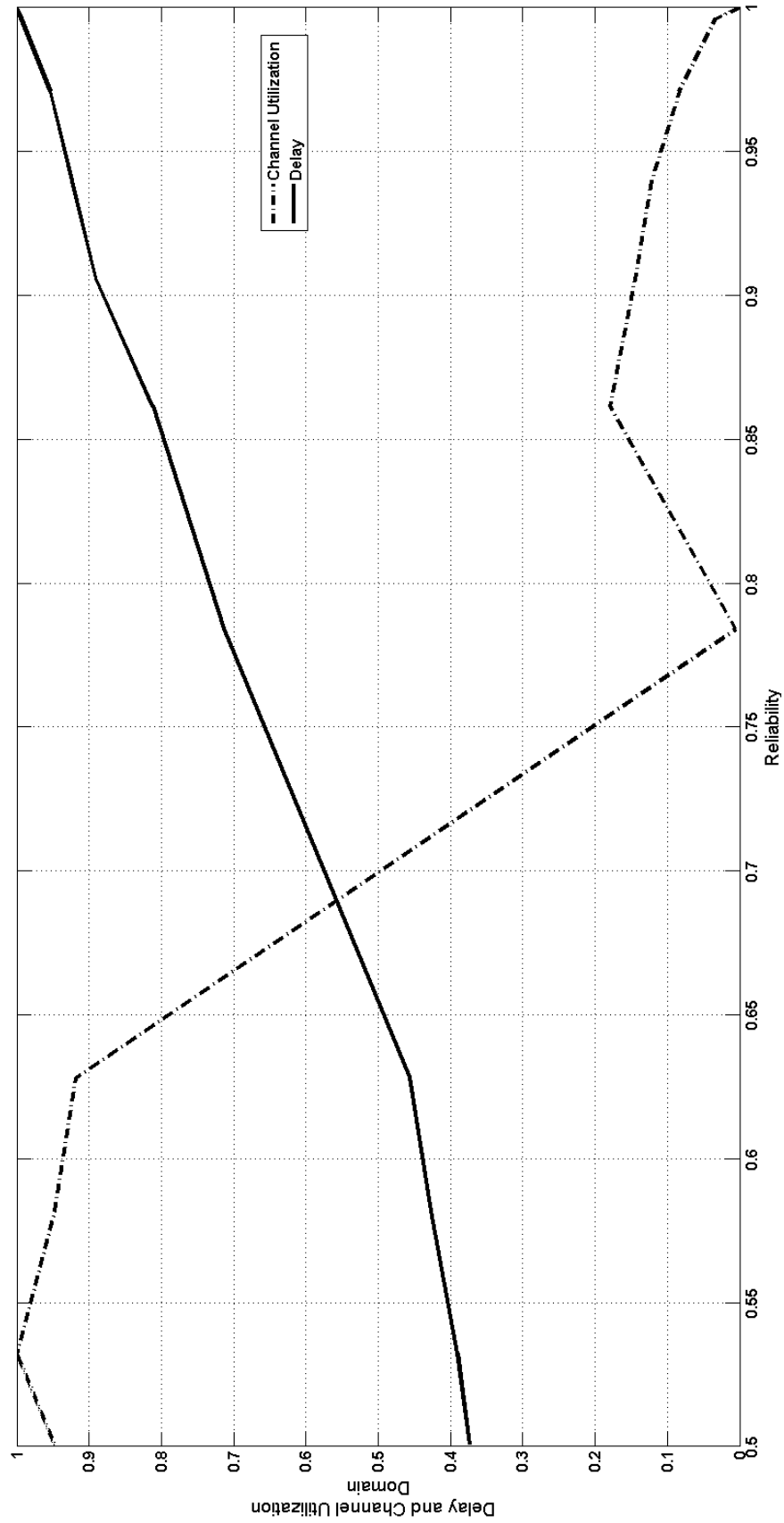


Figure 57. Reliability versus domain of delay and channel utilization for two sensors using ENSURE 2.0

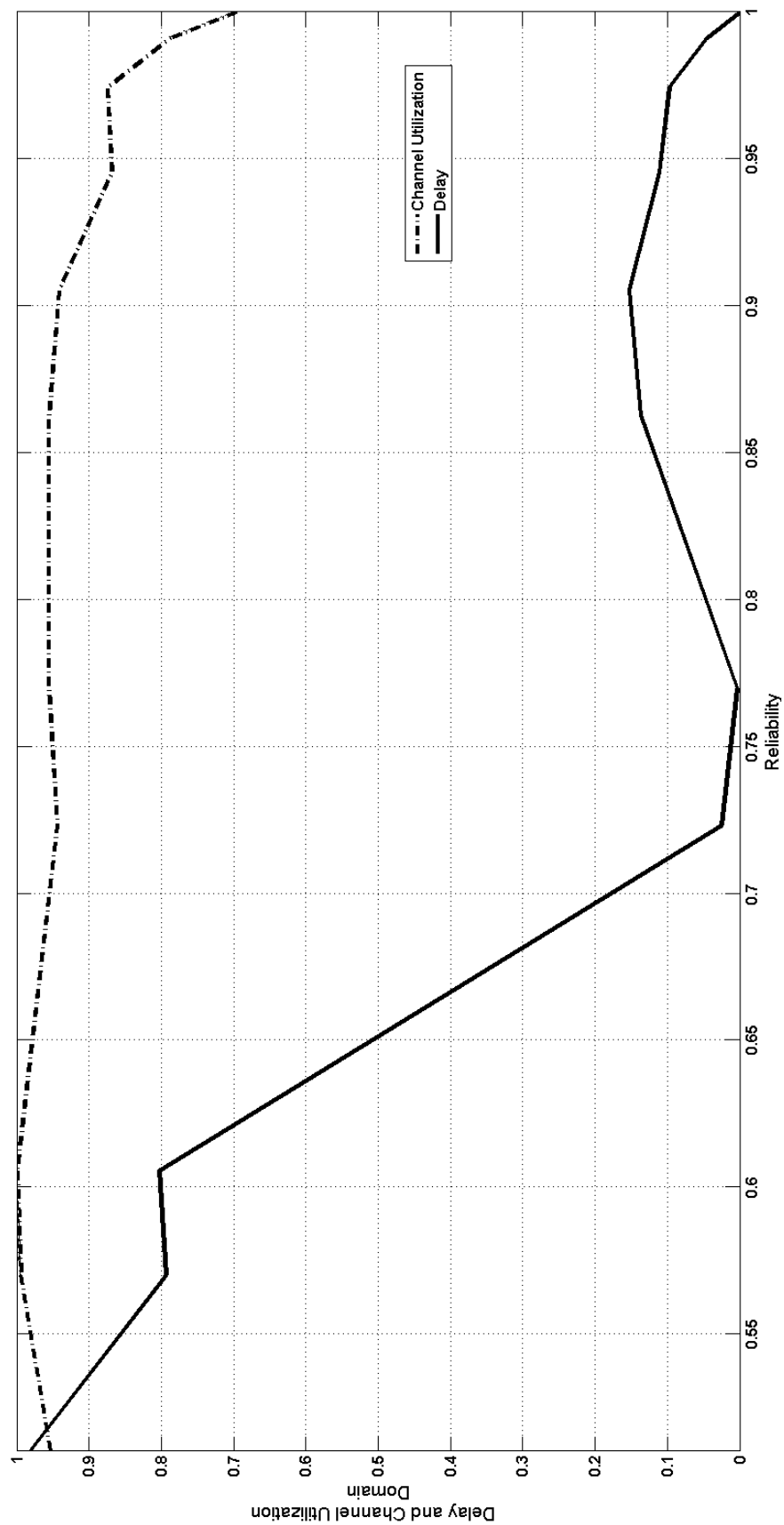


Figure 58. Reliability versus domain of delay and channel utilization for three sensors using ENSURE 2.0

5.4 ENSURE 3.0: Error Correction

The petrochemical industry has two very critical needs out of a wireless network, reliability and channel utilization. Each of these requirements has been met individually by the two previous versions of ENSURE. ENSURE 1.0 proved that the protocol could guarantee packet reception while 2.0 provided reasonable time delays. Now the protocol must combine the two characteristics to provide a fast, reliable means of reading transmissions, thereby creating ENSURE 3.0.

The final version of the protocol will combine the best of both worlds by adding an error correction mechanism. For the present simulation, a robust cyclic error-correcting code, Reed Solomon, was chosen. This is because it has the ability to correct erasures by filling in missing blocks of code. In channel conditions that were simulated in the ENSURE networks Reed Solomon was capable of recovering 190 erasures. By adding this code to packets sent over the ENSURE 2.0, the final 3.0 protocol took shape. All channel conditions were recorded allowing the effectiveness of the transmission parameters to be examined.

5.4.1 Delay versus Channel Utilization

In order to satisfy industrial needs, the final protocol must be able to maintain a minimal total delay with maximum channel utilization. With the combination, plant networks can be designed around the current equipment to effectively use the resources required to connect the PLC to the base station. With the changes that were made in the packet coding process, one would expect to see an increase in channel utilization, but it would still not be able to obtain total usage. In ENSURE 3.0 simulations running one sensor, Figure 56, shows no improvement over the previous setup up, however, the difference is noticeable with two reading sources, Figure 57. There is only a 10% drop in usage when the delay is initially added and the overall range of usage is 30% to 50%.

Keeping with this pattern, when three sensors are present (Figure 58), channel conditions are favorable with a maximum utilization of approximately 55%, a minimum of 30%, and a steady downward slope from 0.05 to 4.5 seconds delay. Figure 59 shows that three sensors will make the most of available resources while not compromising delay when using ENSURE 3.0.

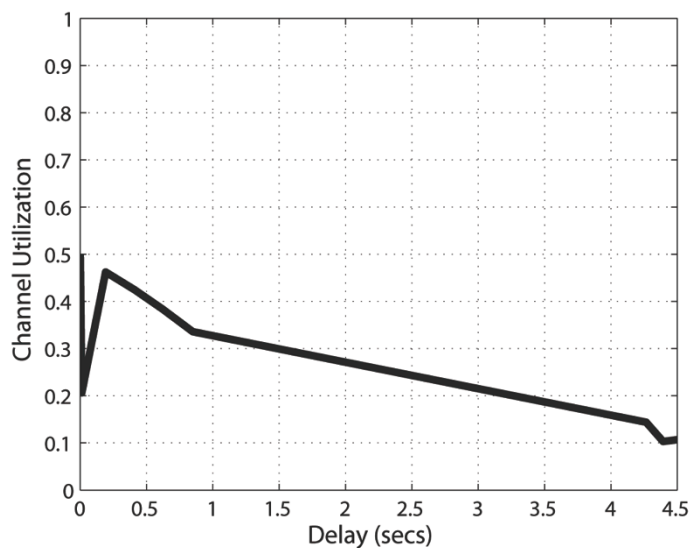


Figure 59. Delay versus channel utilization for one sensor using ENSURE 3.0

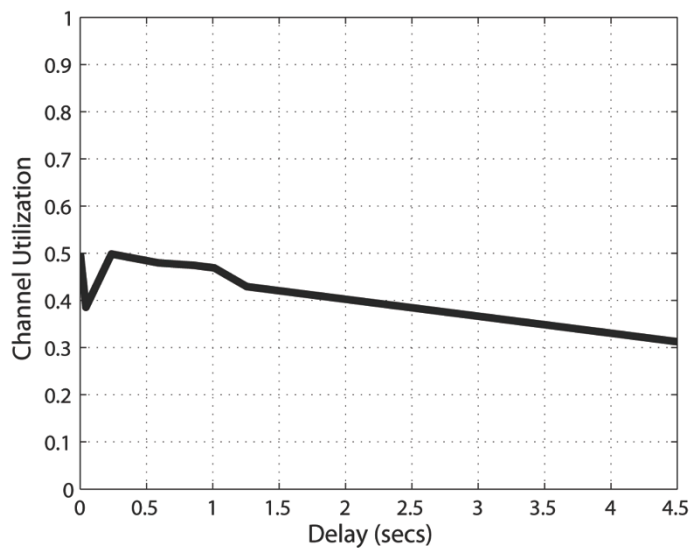


Figure 60. Delay versus channel utilization for two sensors using ENSURE 3.0

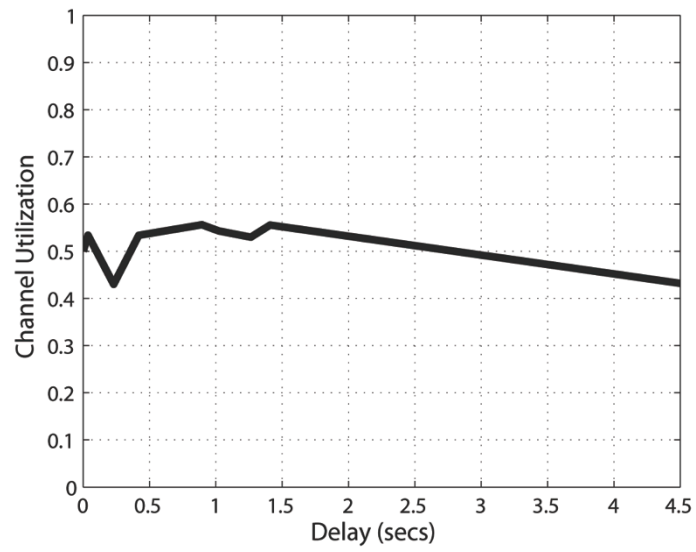


Figure 61. Delay versus channel utilization for three sensors using ENSURE 3.0

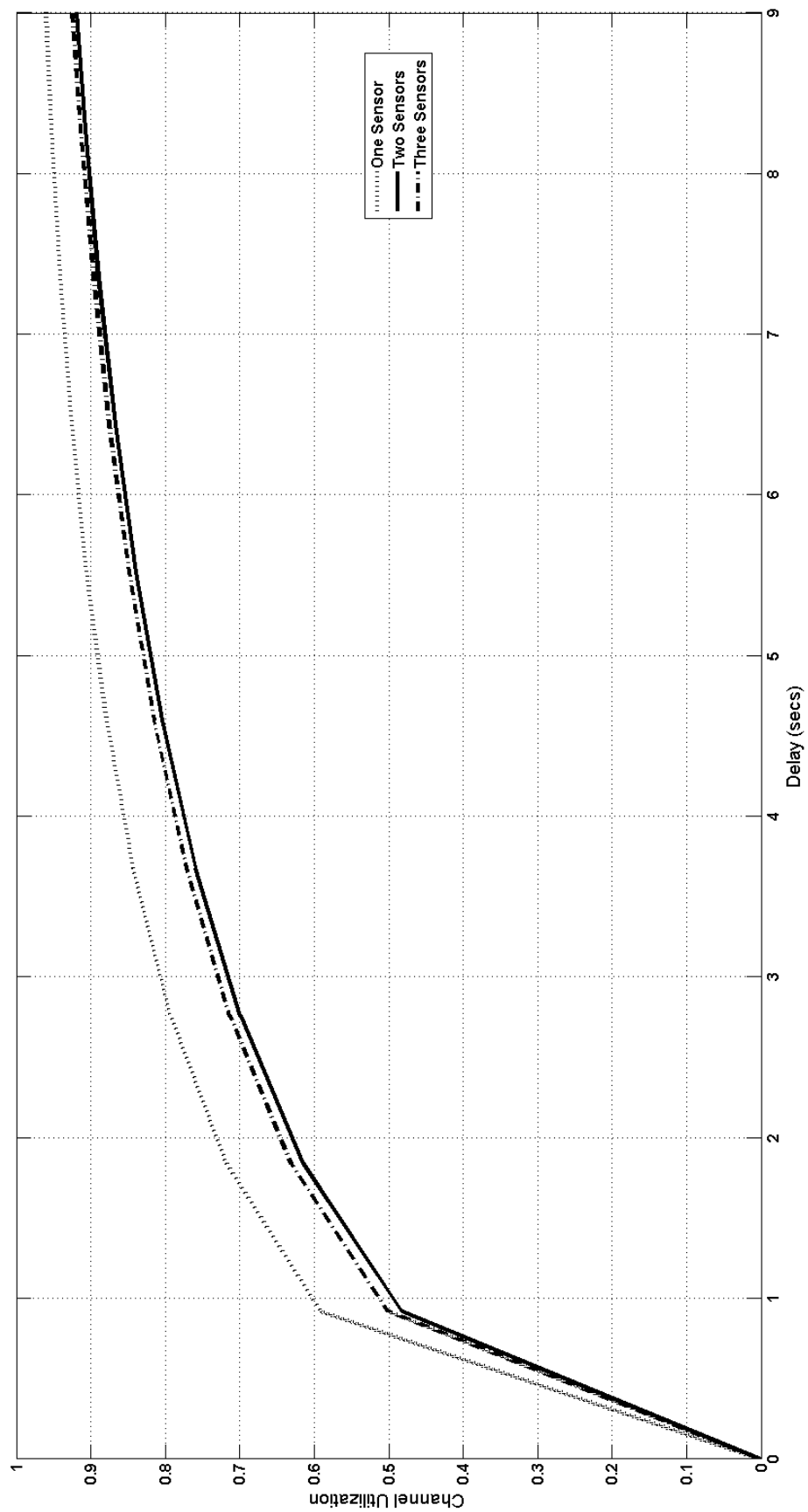


Figure 62. Comparison delay versus channel utilization for all sensors using ENSURE 3.0

5.4.2 Noise versus Channel Utilization

As with the all of the ENSURE protocols, 3.0 has a floating transmission window which reacts to current reliability of the network. When a time limit was factored in before with one sensor, no losses occurred, but developed in the two and three sensor runs. As such, Figure 60 is exactly the same as before, whereas Figure 61 shows the range of channel utilization to be steady but the reaction to noise to be different. Decreases in usage are not detected until a 25% noise level, and a major decrease is seen at 35% before leveling out to a slight, steady decrease continuing through 50% noise levels. The utilization with three sources, Figure 62, is much the same, but there is only one range of noise, 30% to 40%, that has a sharp decrease. With these graphs, along with Figure 63, one can conclude that the protocol has indeed effectively minimized the effects of noise on the base station's ability to decode data.

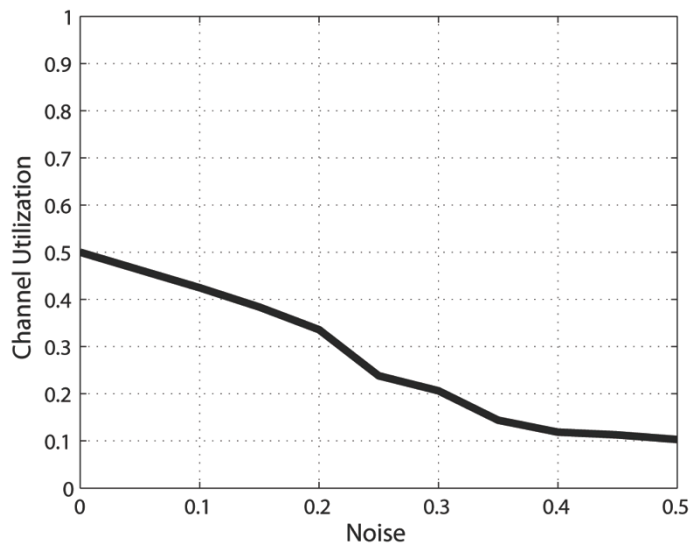


Figure 63. Noise versus channel utilization for one sensor using ENSURE 3.0

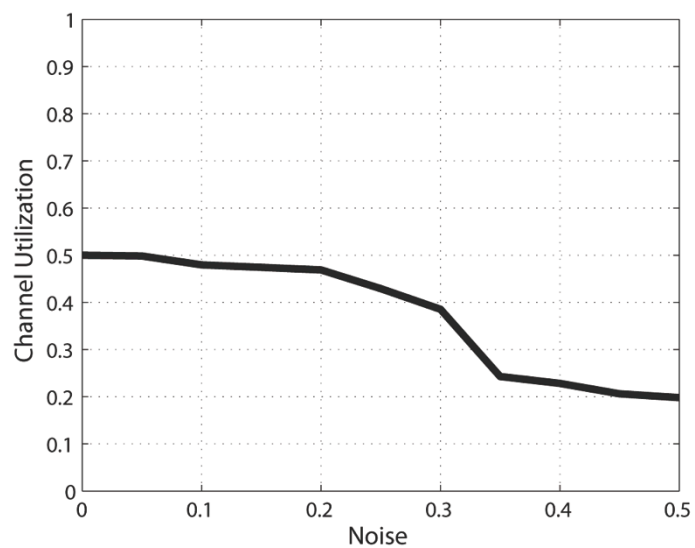


Figure 64. Noise versus channel utilization for two sensors using ENSURE 3.0

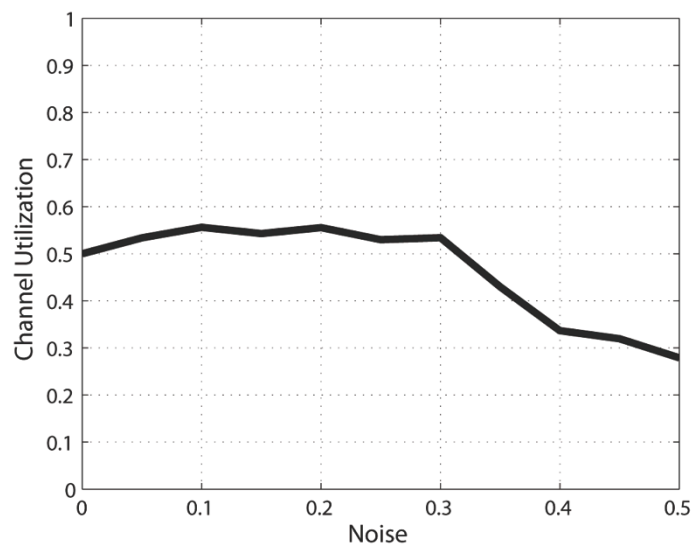


Figure 65. Noise versus channel utilization for three sensors using ENSURE 3.0

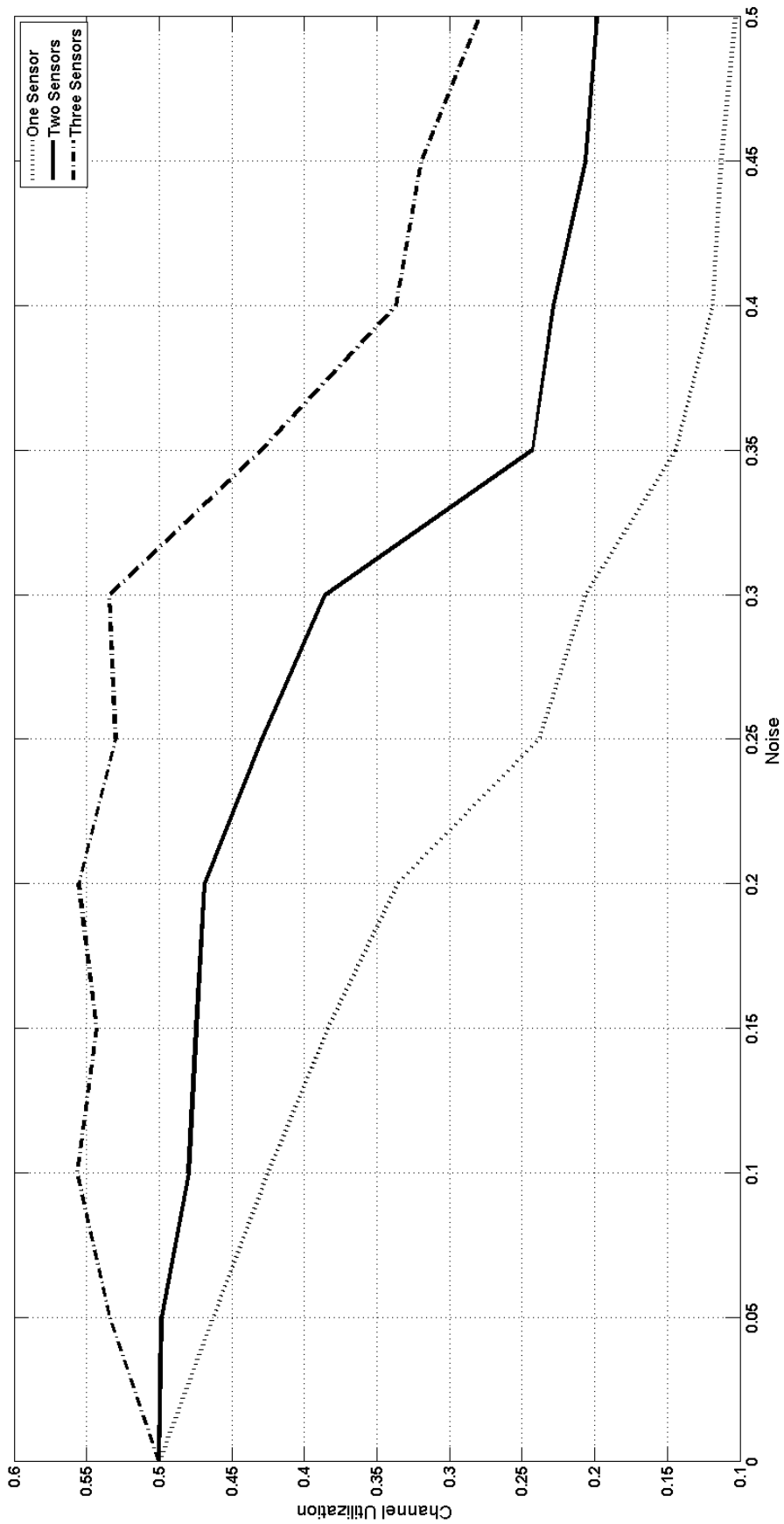


Figure 66. Comparison noise versus channel utilization for all sensors using ENSURE 3.0

5.4.3 Channel Utilization versus Reliability

Wireless networks can often be thought of a series of algorithms that produce measures which a network engineer must determine the weights of to produce an equilibrium that meets their end goals. ENSURE was created with a goal of creating 100% reliability with a reasonable amount of channel utilization, and 3.0 finds the unique balance that meets these requirements. The best demonstration of the effectiveness is to show channel utilization, noise levels, and reliability side by side. All three configurations, Tables 7 – 9, show that no packets were lost and the channel had a tolerable amount of usage when the Reed Solomon coding was added.

Table 7. Channel utilization versus reliability for one sensor using ENSURE 3.0.

One Sensor		
Channel Utilization	Noise	Reliability
0.50	0	1
0.46	0.05	1
0.42	0.1	1
0.38	0.15	1
0.34	0.2	1
0.24	0.25	1
0.21	0.3	1
0.14	0.35	1
0.12	0.4	1
0.11	0.45	1
0.10	0.5	1

Table 8. Channel utilization versus reliability for two sensors using ENSURE 3.0

Two Sensors		
Channel Utilization	Noise	Reliability
0.50	0	1
0.50	0.05	1
0.48	0.1	1
0.47	0.15	1
0.47	0.2	1
0.43	0.25	1
0.39	0.3	1
0.24	0.35	1
0.23	0.4	1
0.21	0.45	1
0.20	0.5	1

Table 9. Channel utilization versus reliability for three sensors using ENSURE 3.0

Three Sensors		
Channel Utilization	Noise	Reliability
0.50	0	1
0.53	0.05	1
0.56	0.1	1
0.54	0.15	1
0.56	0.2	1
0.53	0.25	1
0.53	0.3	1
0.43	0.35	1
0.34	0.4	1
0.32	0.45	1
0.28	0.5	1

5.4.4 Noise versus Retransmissions

Since ENSURE 3.0 aims to combine its two predecessors, one would expect that the retransmissions measurements to reflect balance of the two prior limitations. Again,

since no loss was detected in a one-sensor network in either prior protocol configuration, Figure 64 is an exact reflection of ENSURE 2.0's results. With an additional sensor, Figure 65, the simulation results clearly show the positive effects of the Reed Solomon coding. Retransmissions were kept to a minimum until a high level of noise, 35%, was reached, and then the reliability of the channel begins to cause drastic adjustments to the number sent. The same holds true when simulating three reading points, Figure 66, with a clear change taking place at 45%. The reason that the adjustment is made later is that the initial window is larger and when a single packet loss is detected it has less effect on the network reliability. This is a measure to help determine an ideal setup based on noise. Figure 67 demonstrates that different level ranges shift how many sensors are ideal.

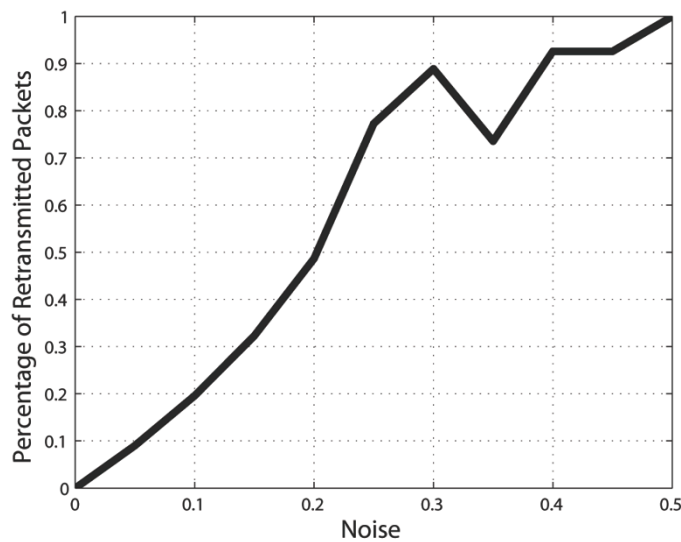


Figure 67. Noise versus retransmissions for one sensor using ENSURE 3.0

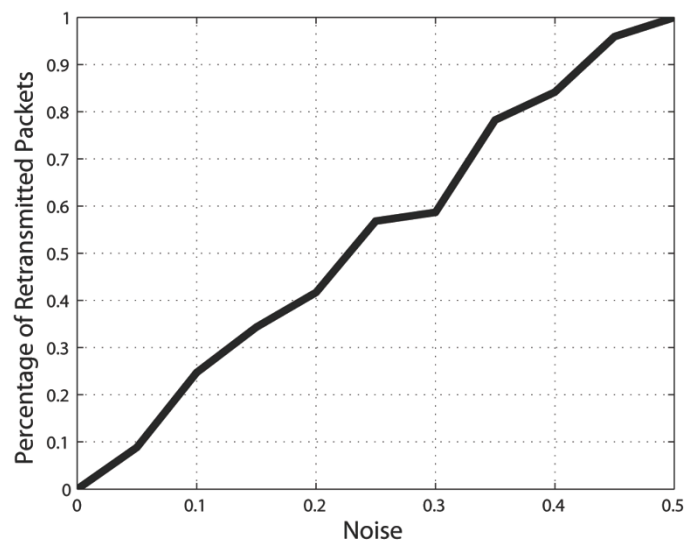


Figure 68.Noise versus retransmissions for two sensors using ENSURE 3.0

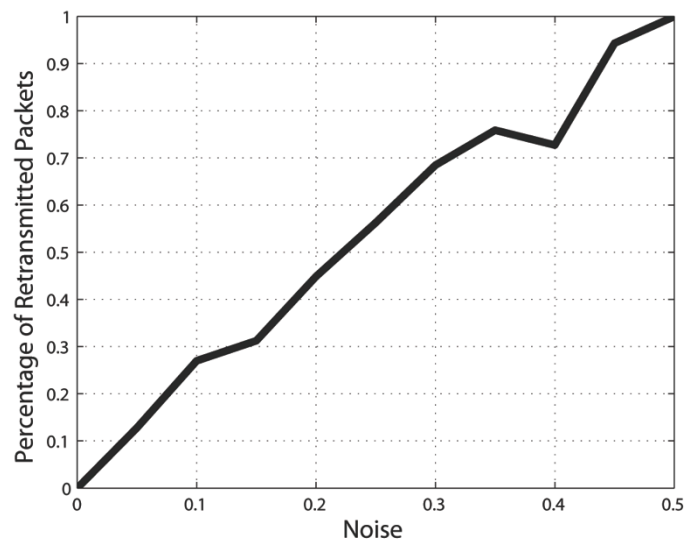


Figure 69. Noise versus retransmissions for three sensors using ENSURE 3.0

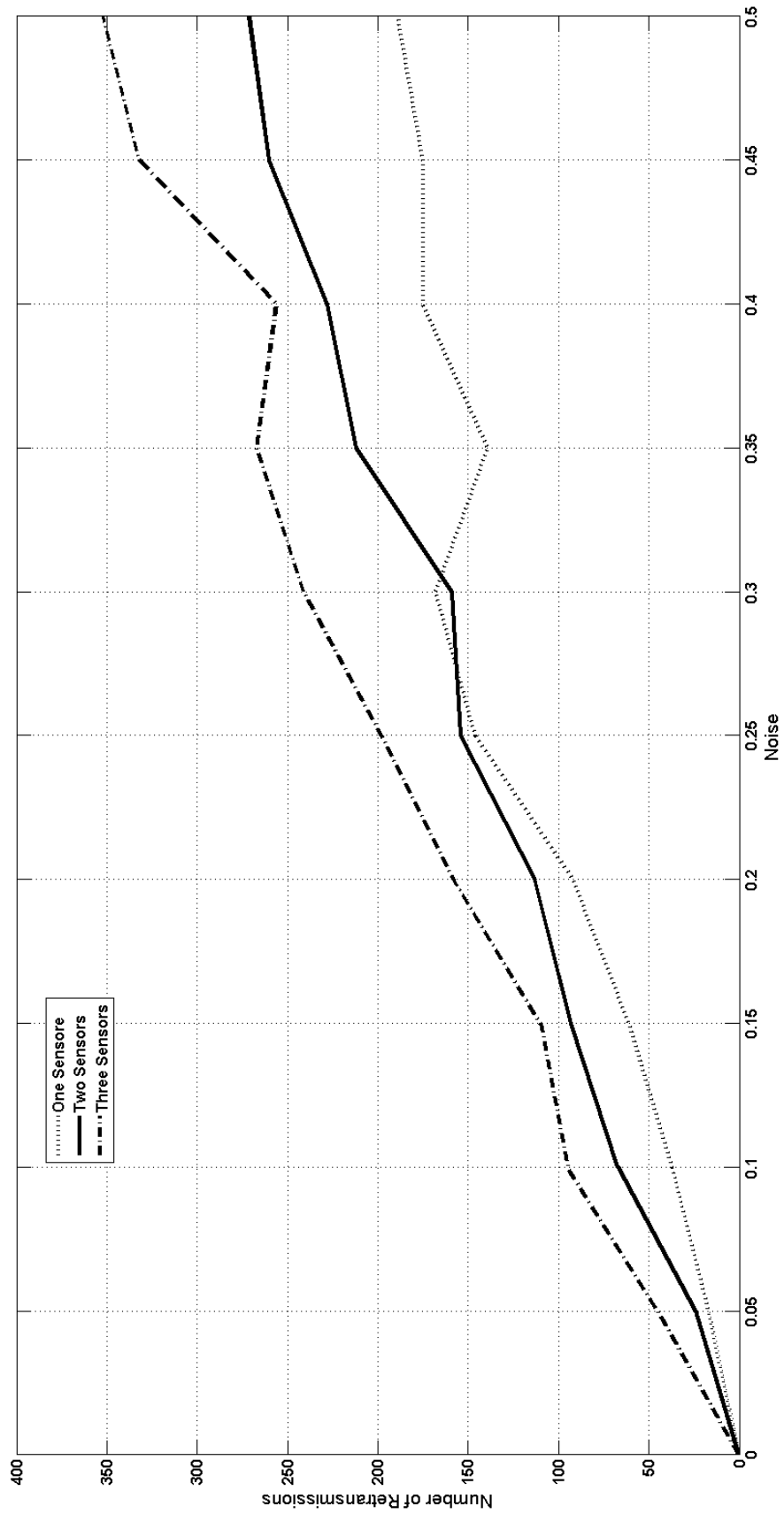


Figure 70. Comparison of noise versus retransmissions for all sensors using ENSURE 3.0

5.4.5 Reliability versus Retransmissions

Measuring the reliability versus retransmissions in this protocol serves to reinforce the adjustments the network makes in the event of a loss is effective. As seen in Tables 10 - 12, the number of sensors and noise level both contribute to determine the number of retransmissions required for a successful packet decoding.

Table 10. Reliability versus retransmissions for one sensor using ENSURE 3.0

One Sensor		
Retransmissions	Reliability	Noise
498	1	0
532	1	0.05
572	1	0.1
620	1	0.15
682	1	0.2
932	1	0.25
998	1	0.3
1300	1	0.35
510	1	0.4
1532	1	0.45
1496	1	0.5

Table 11. Reliability versus retransmissions for two sensors using ENSURE 3.0

Two Sensors		
Retransmissions	Reliability	Noise
996	1	0
1046	1	0.05
1136	1	0.1
1212	1	0.15
1284	1	0.2
1379	1	0.25
1603	1	0.3
2424	1	0.35
2544	1	0.4
2694	1	0.45
2664	1	0.5

Table 12. Reliability versus retransmissions for three sensors using ENSURE 3.0

Three Sensors		
Retransmissions	Reliability	Noise
1494	1	0
1598	1	0.05
1722	1	0.1
1772	1	0.15
1914	1	0.2
2016	1	0.25
2414	1	0.3
2540	1	0.35
3383	1	0.4
3523	1	0.45
3852	1	0.5

5.4.6 Noise versus Packets Loss

As was the case with ENSURE 1.0, comparing the amount of noise against the number of lost packets reinforces the validity of previous measures. Tables 10 - 12 in 5.4.5 contain supporting evidence that a packet was never lost in any of the simulations.

5.4.7 Noise versus Domain of Delay and Channel Utilization

Once again, plotting noise level, length of delay and channel usage into one graph will prove useful. The outcomes for ENSURE 3.0 simulations show clear cutoff points of desirable noise levels. The plots of delay and usage still cross, however, after that point the plots mirror each other using the intersection point as the relative axis. When comparing at Figures 68 - 70, one can conclude that when a sensor is added the intersection point shifts. The importance of this is the balance point has been discovered, offering plants a benchmark to use when implementing a network.

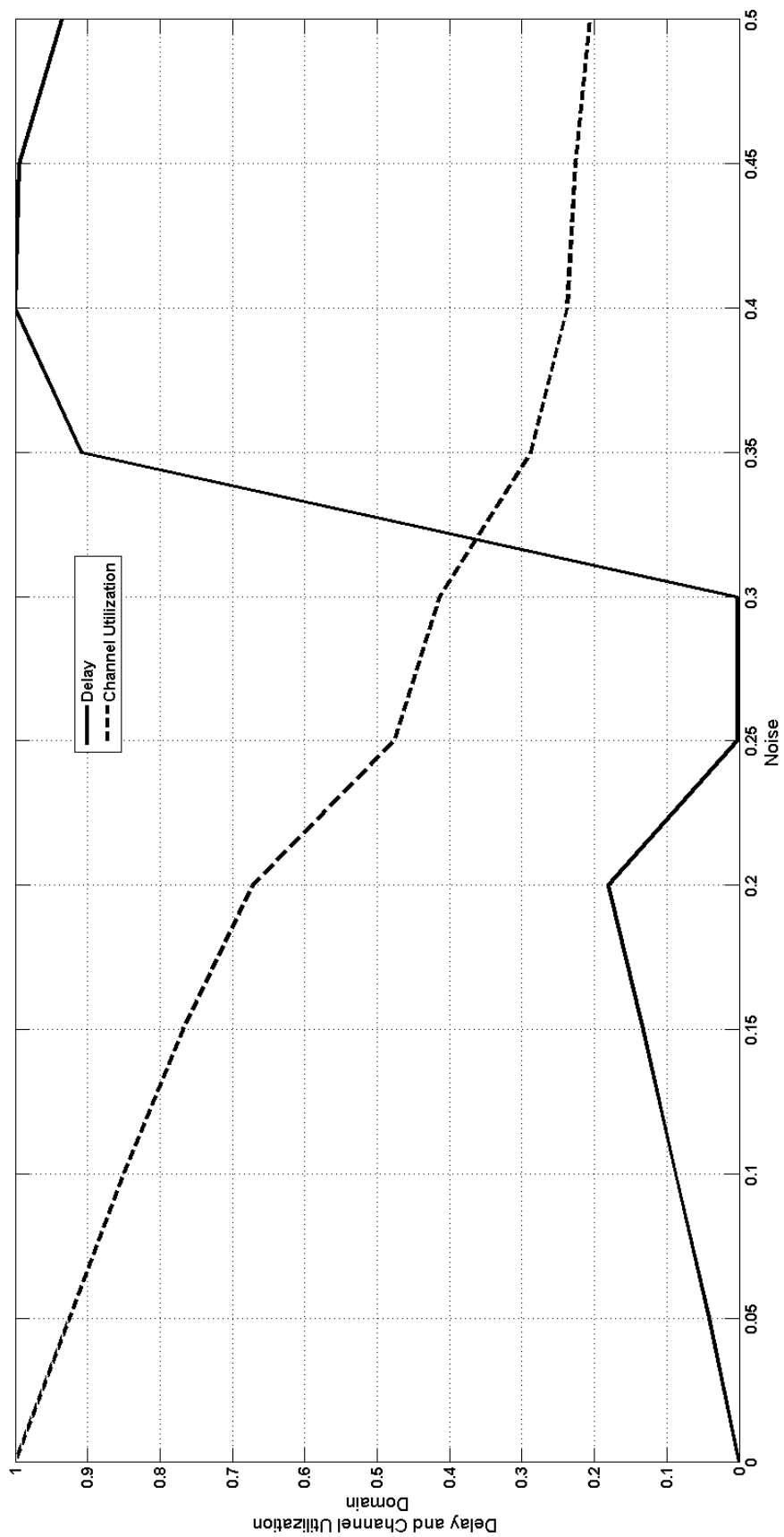


Figure 71. Noise versus domain of delay and channel utilization for one sensor using ENSURE 3.0

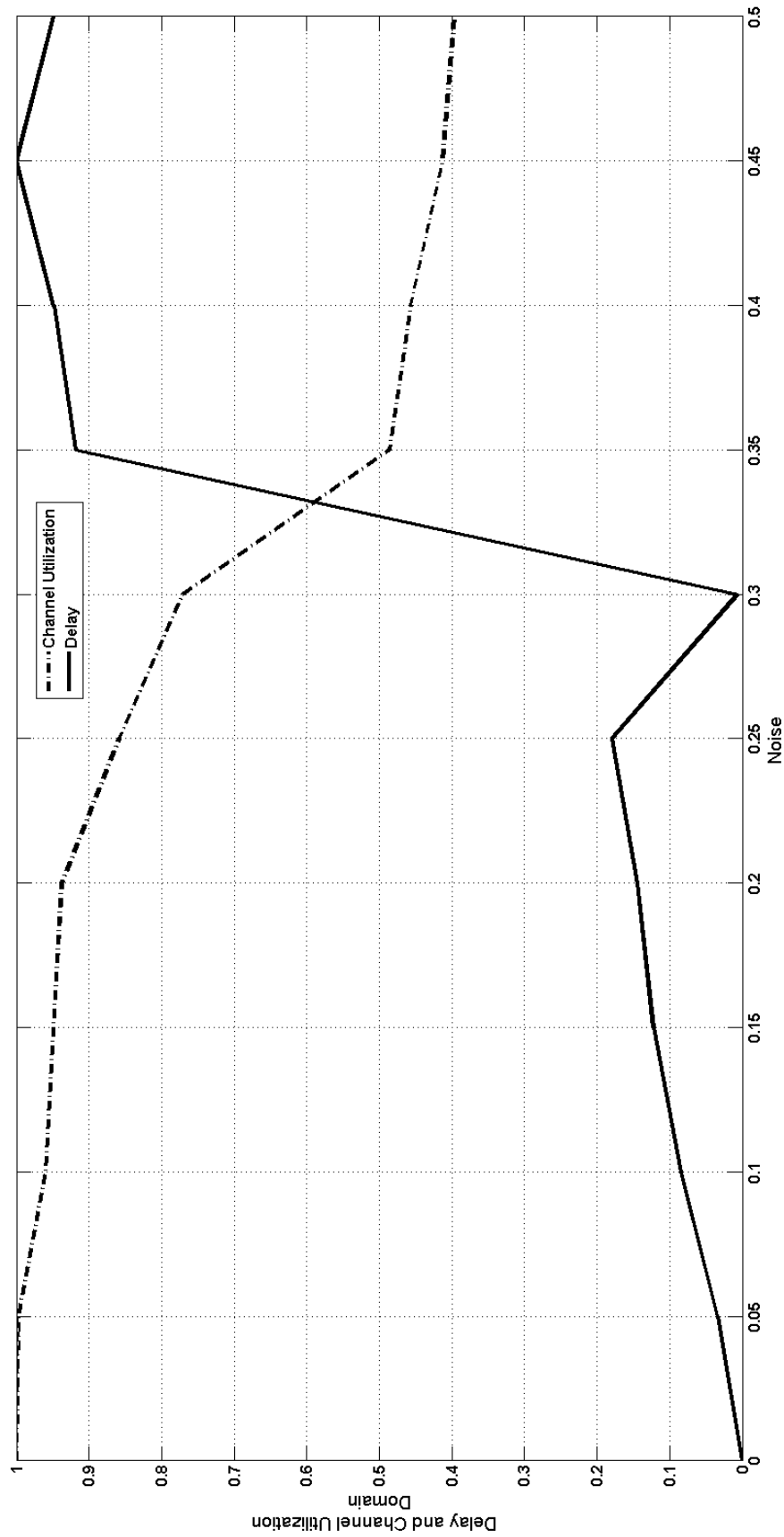


Figure 72. Noise versus domain of delay and channel utilization for two sensors using ENSURE 3.0

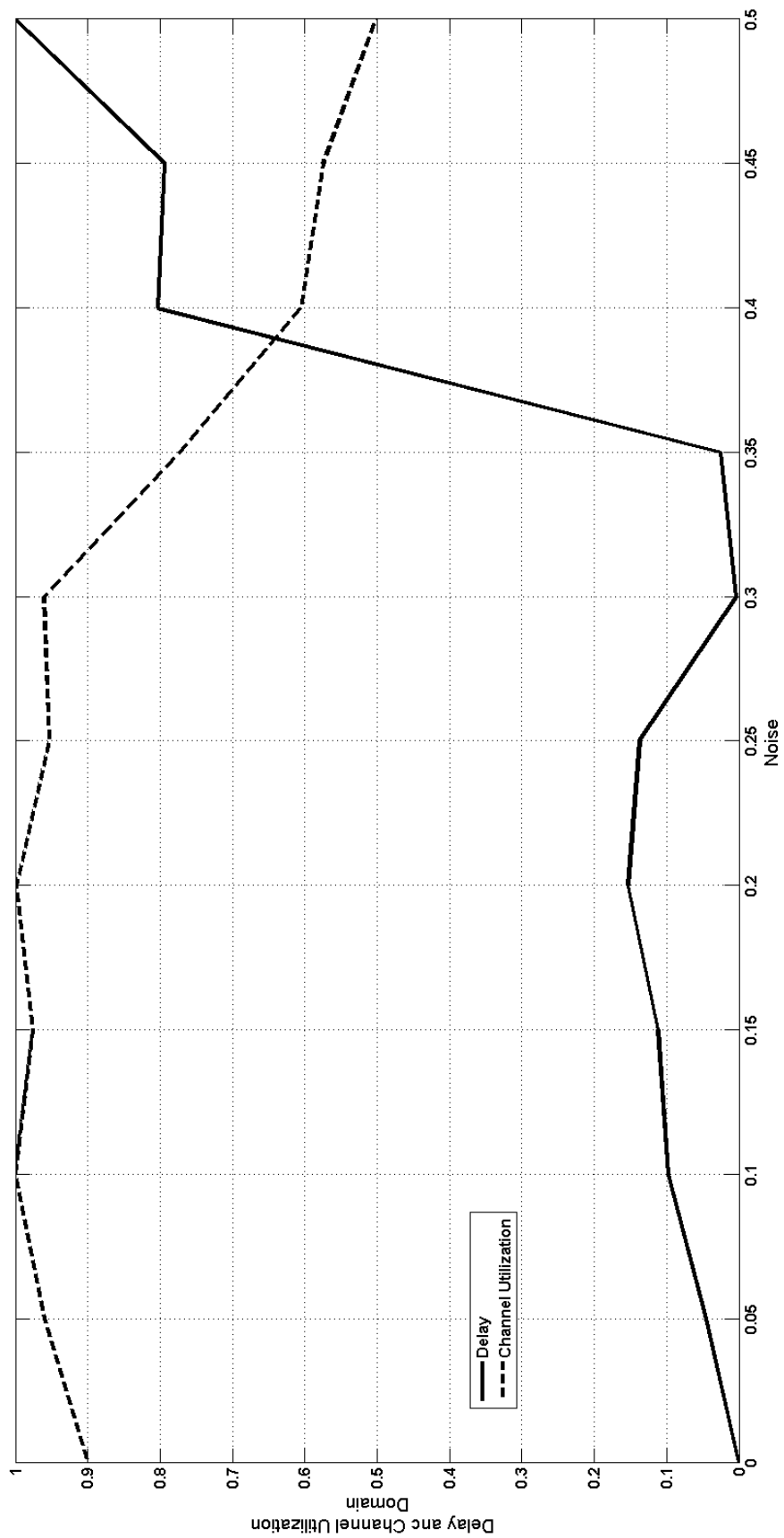


Figure 73. Noise versus domain of delay and channel utilization for three sensors using ENSURE 3.0

5.4.8 Reliability versus Domain of Delay and Channel Utilization

The measurements for reliability versus delay and channel utilization reflect the same patterns of the results for noise. The take-a-way is that the lines cross at the same levels as the previous sections. This validates the points previously made.

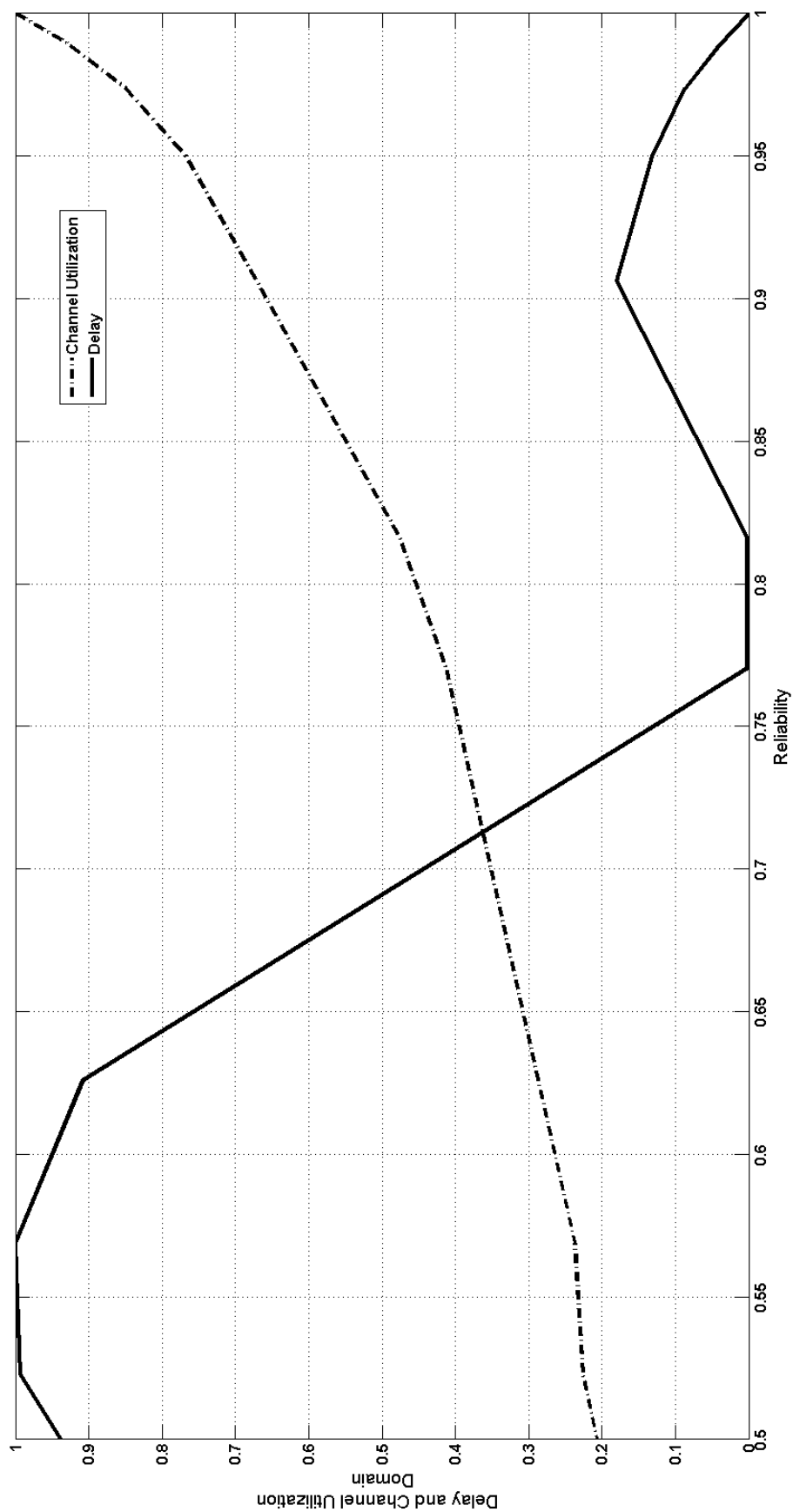


Figure 74. Reliability versus domain of delay and channel utilization for one sensor using ENSURE 3.0

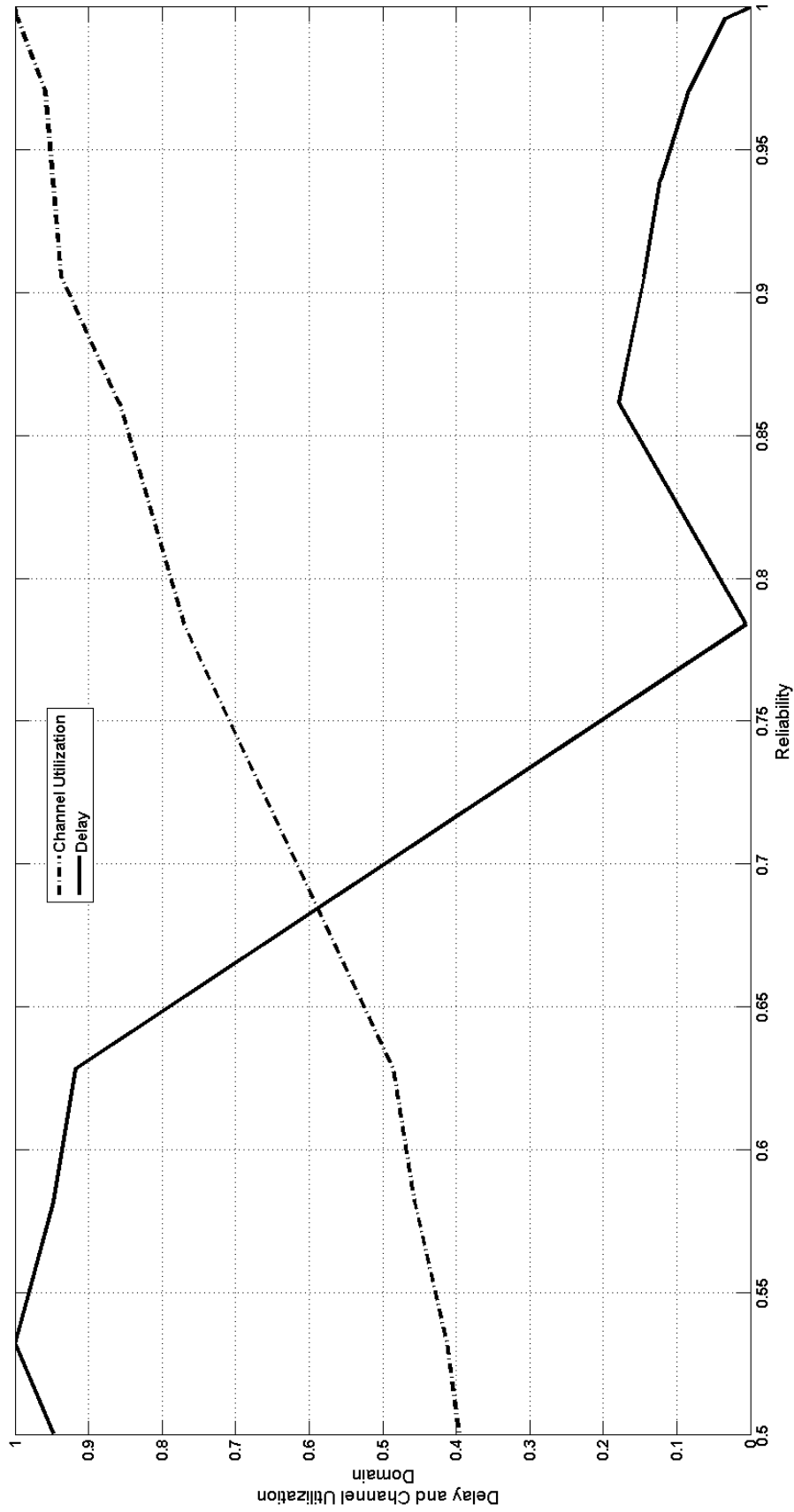


Figure 75. Reliability versus domain of delay and channel utilization for two sensors using ENSURE 3.0

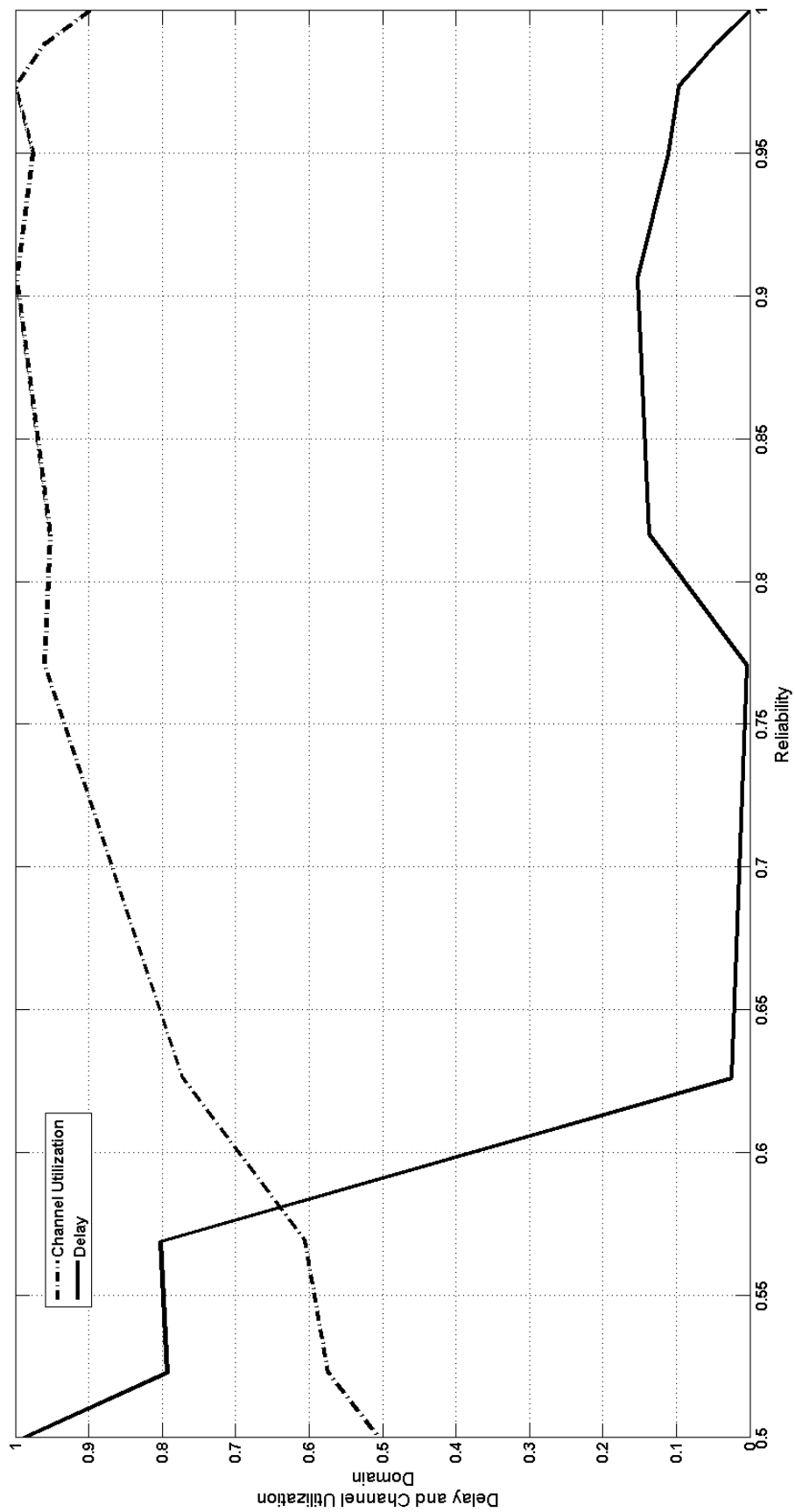


Figure 76. Reliability versus domain of delay and channel utilization for three sensors using ENSURE 3.0

5.5 Overall Transport Protocol Improvements

With each adjustment to ENSURE, improvements were made that resulted in a final protocol that met the goal of providing 100% reliability while maintaining acceptable channel usage under a restricted time limit. However, ultimately one must look at the overall picture and ask: Does this improve on the standards that are in place today? A good base line to rank it against is the RDT 2.0 protocol.

To examine the performance differences a Simulink model of RDT 2.0 was developed and run under the same conditions as the ENSURE protocols. Again, at the end of each simulation statistics pertaining to the channel were recorded. Then graphs of the key measures were generated that compare each protocol by the number of sensors present.

5.5.1 Delay versus Channel Utilization

In order to produce a detailed graph of the channel's reaction throughout each protocol, the delay times for the simulations were normalized. In Figures 74 – 76, RDT 2.0 has a linear increase in channel utilization, meaning that to correctly receive all packets the delay must be long. While the results for ENSURE 1.0 show steady resource usage, ENSURE 2.0 and 3.0 slowly decrease the amount of unique packets sent across the channel. ENSURE 3.0 shows slightly better results when 2 or more reading sources are present. It is important to note that the overall delay in ENSURE 1.0 is much higher than the later versions, as shown in previous sections of this Chapter.

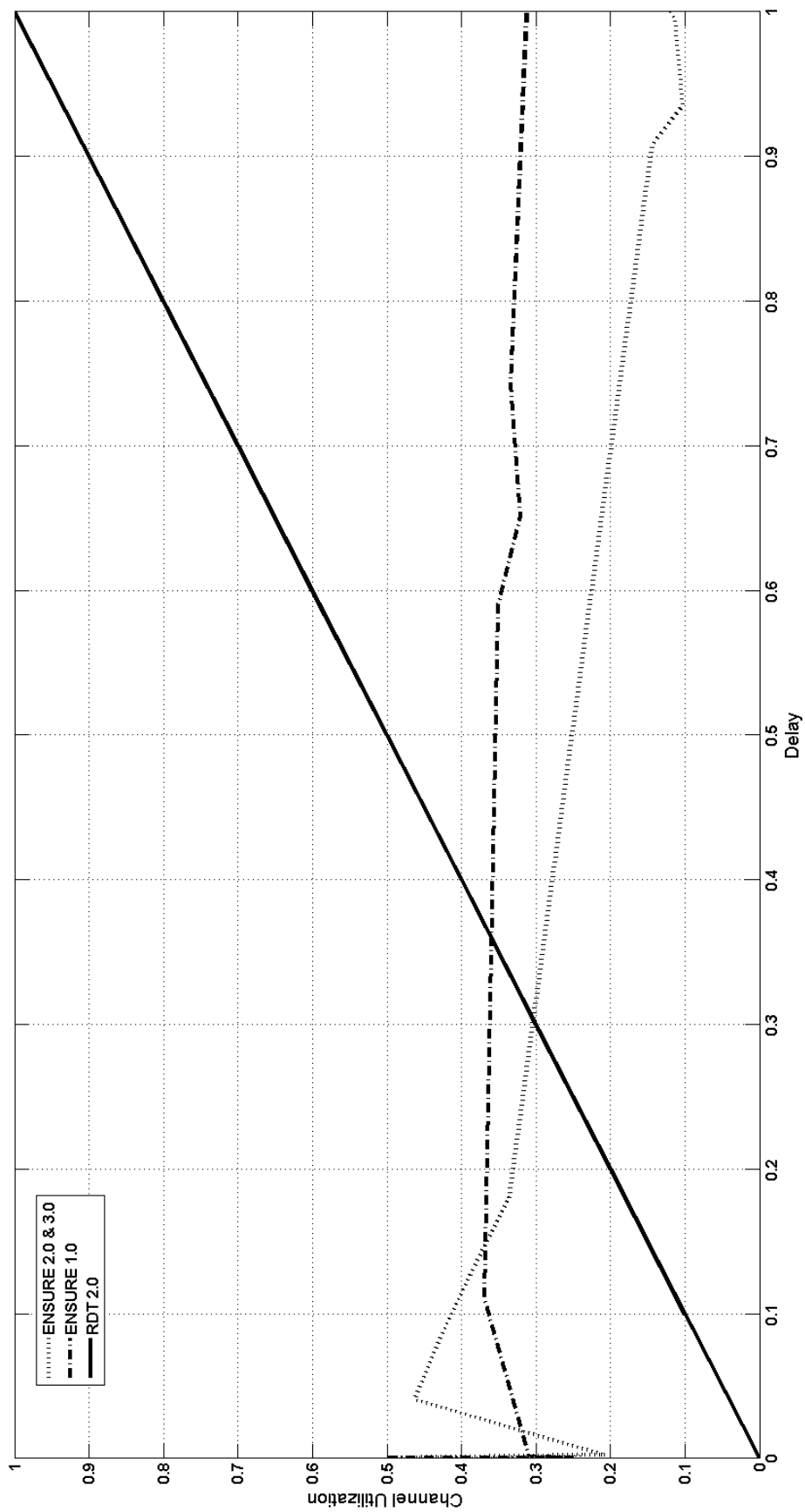


Figure 77. Protocol comparison of delay versus channel utilization for one sensor

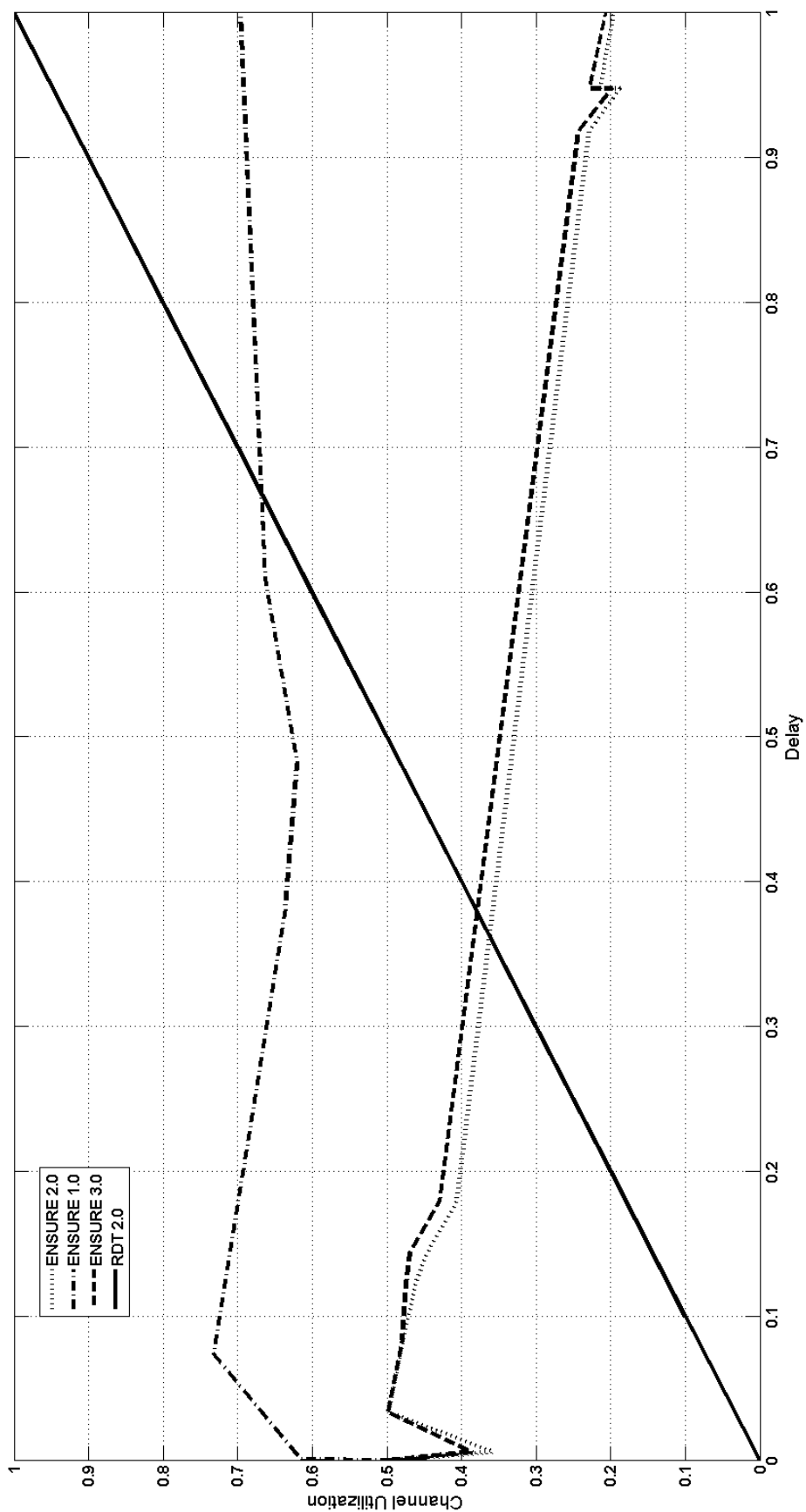


Figure 78. Protocol comparison of delay versus channel utilization for two sensors

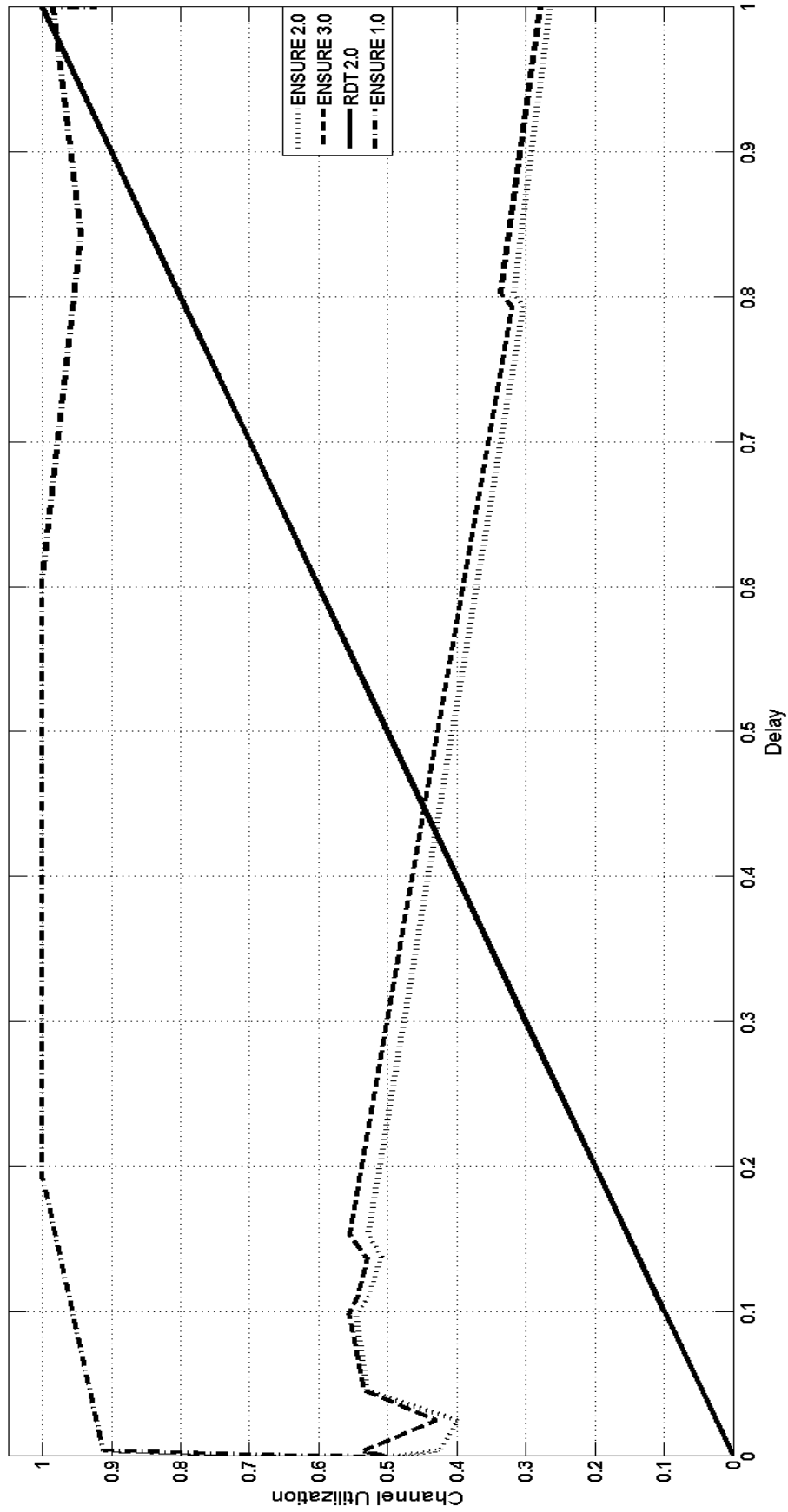


Figure 79. Protocol comparison of delay versus channel utilization for three sensors

5.5.2 Noise versus Channel Utilization

Keeping with the method of normalization, one must examine how the simulation models utilized the channel when noise was added. It can be seen in Figures 77 - 79 that all formats of ENSURE handled noise increases much better than the RDT 2.0 protocol. As more sensors were added to the channel ENSURE 3.0 showed a clear advantage over the other protocols.

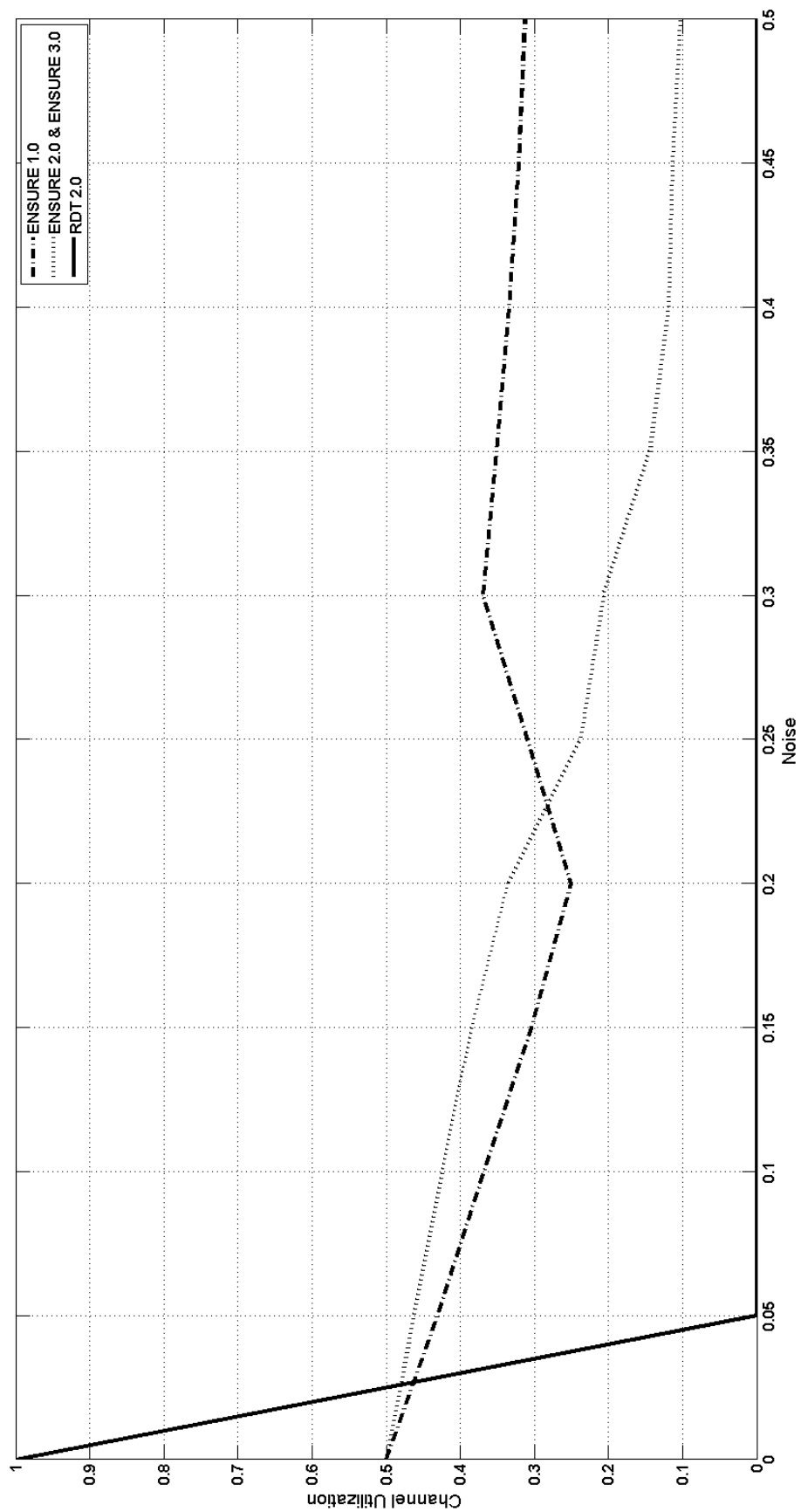


Figure 80. Protocol comparison for noise versus channel utilization of one sensor

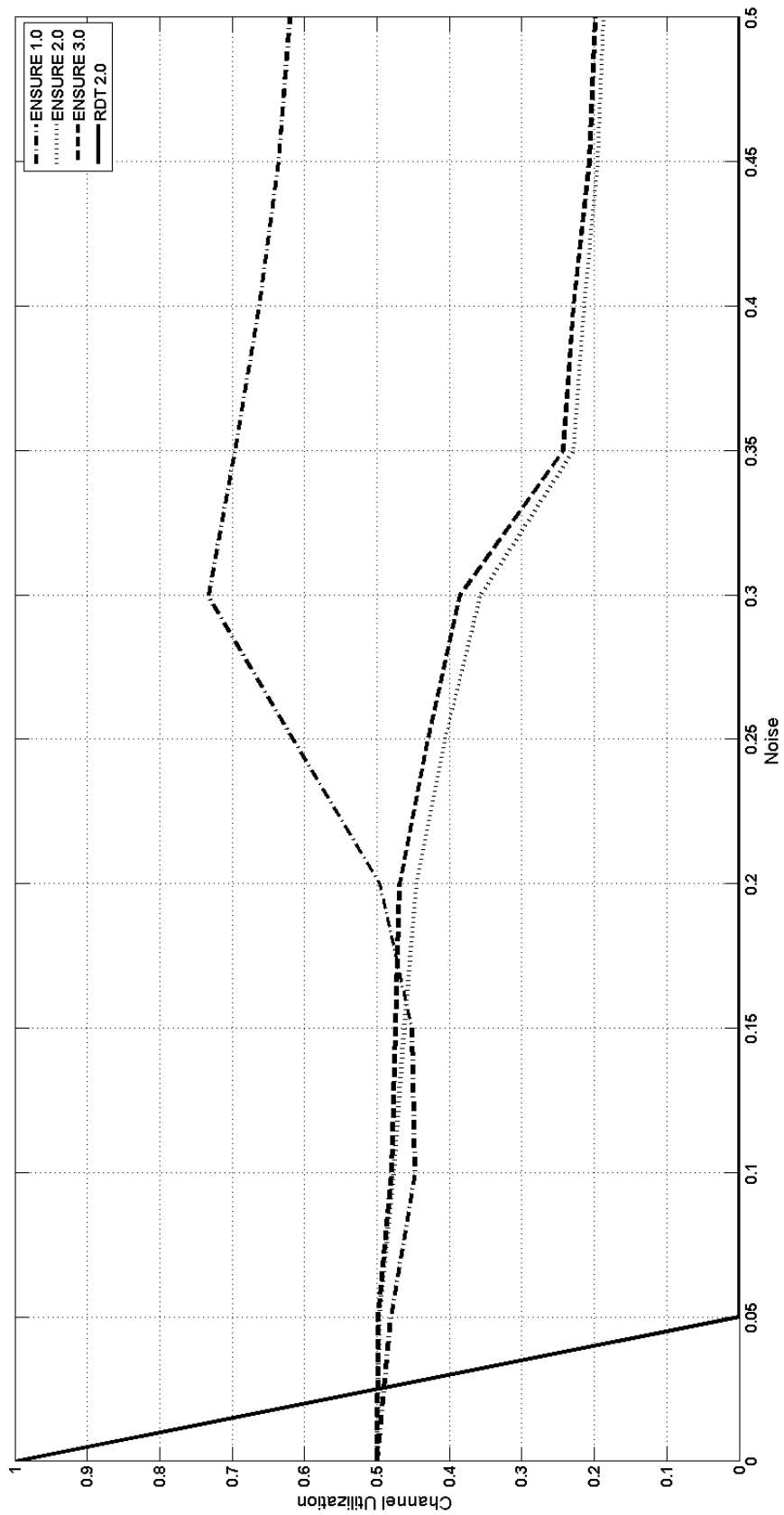


Figure 81. Protocol comparison for noise versus channel utilization of two sensors

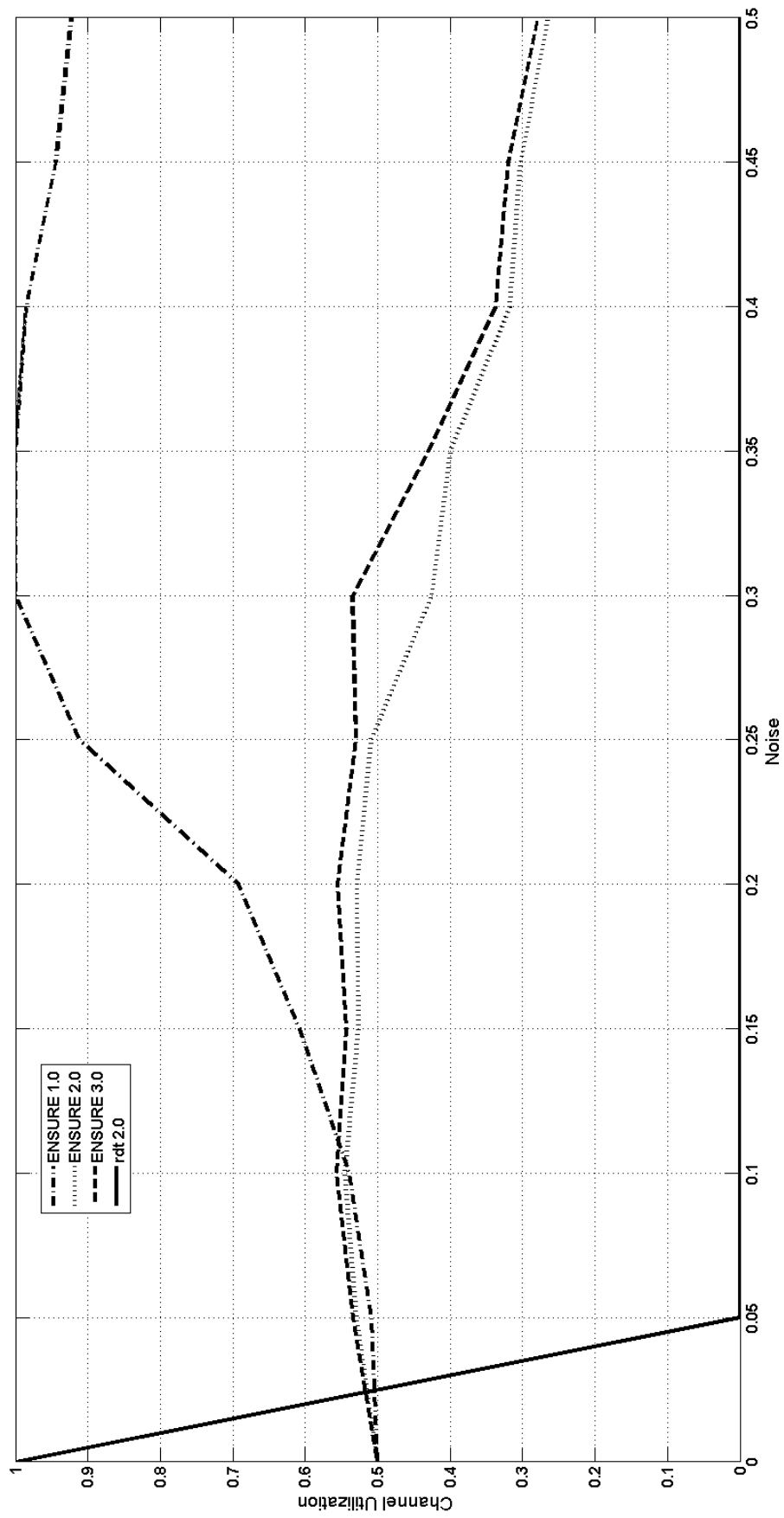


Figure 82. Protocol for noise versus channel utilization of three sensors

5.5.3 Noise versus Packets Loss

A major need when adding a wireless network to petrochemical plants is guaranteed reliability. To place this in a variable that is easy to analyze, it simply means the network cannot lose any packets during the lifetime of the channel. By comparing the number of packets lost in a range of noise levels one can assess the value of the new protocol. RDT 2.0 has extremely high losses when virtually any noise is added (Figures 80 -82), while ENSURE minimizes lost packets. However, with more than one sensor ENSURE 2.0 does produce some missing packets. Versions 1.0 and 2.0 have the ability to successfully transmit all packets across the channel, regardless of network size.

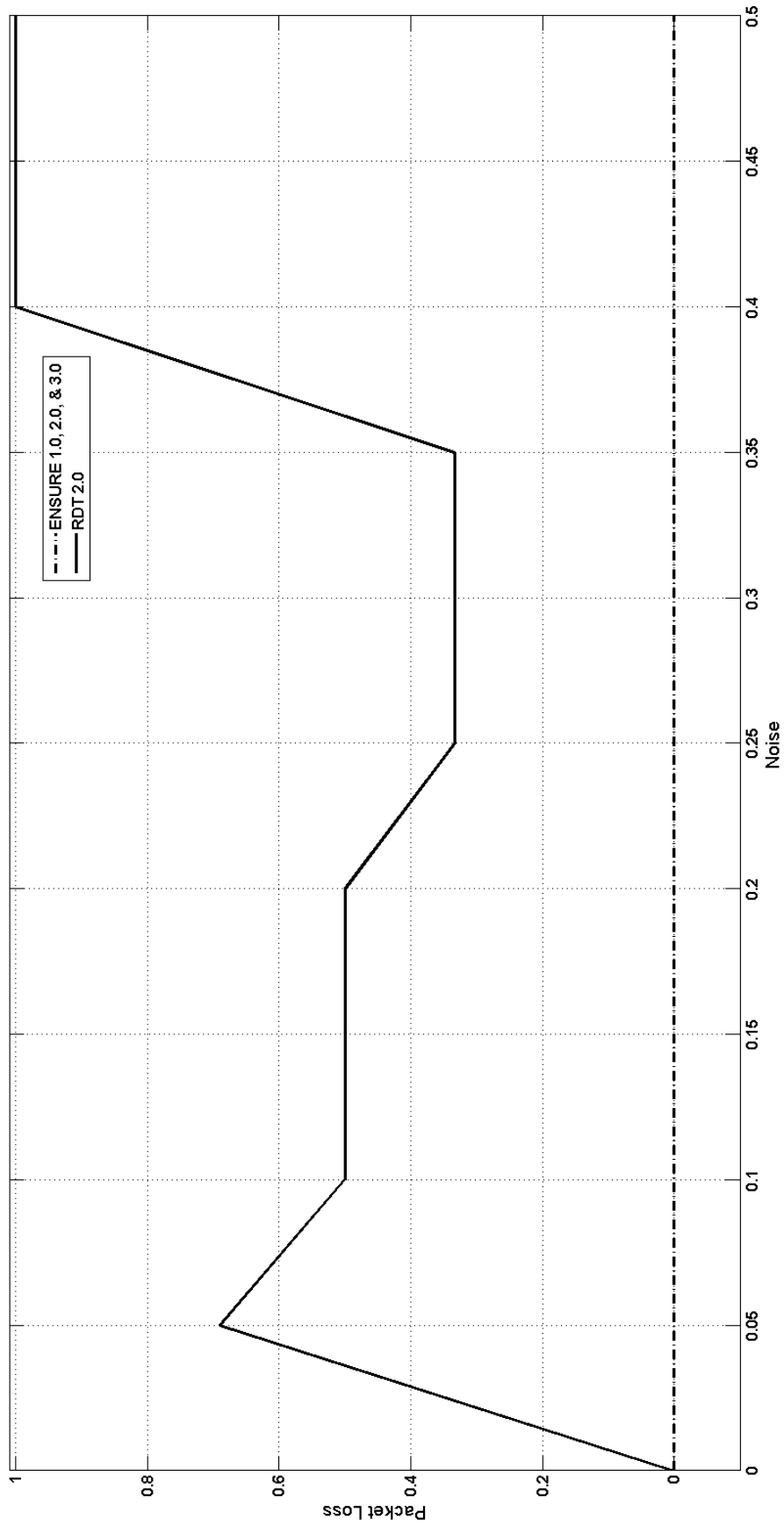


Figure 83. Protocol comparison of noise versus packet loss for one sensor

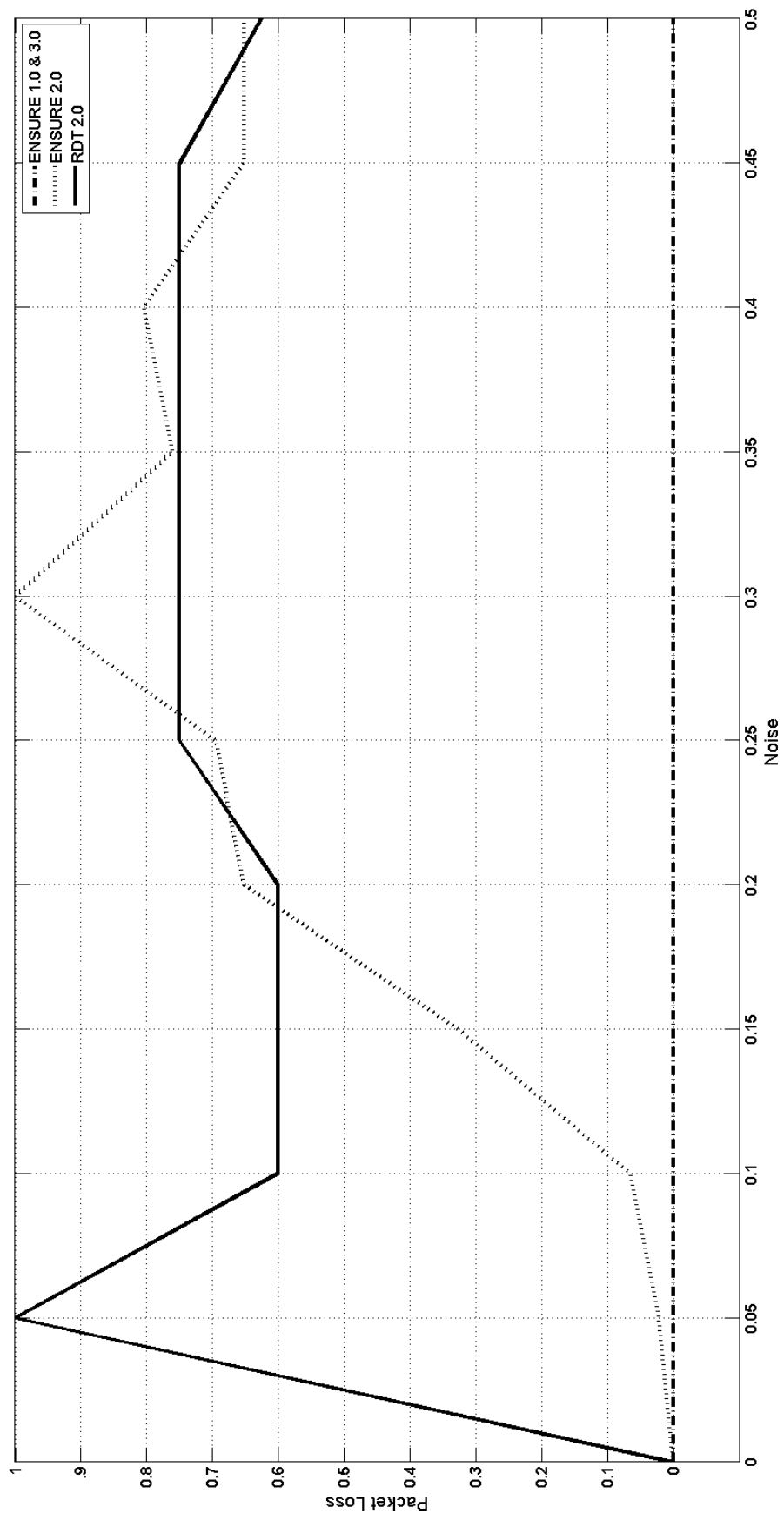


Figure 84. Protocol comparison of noise versus packet loss for two sensors

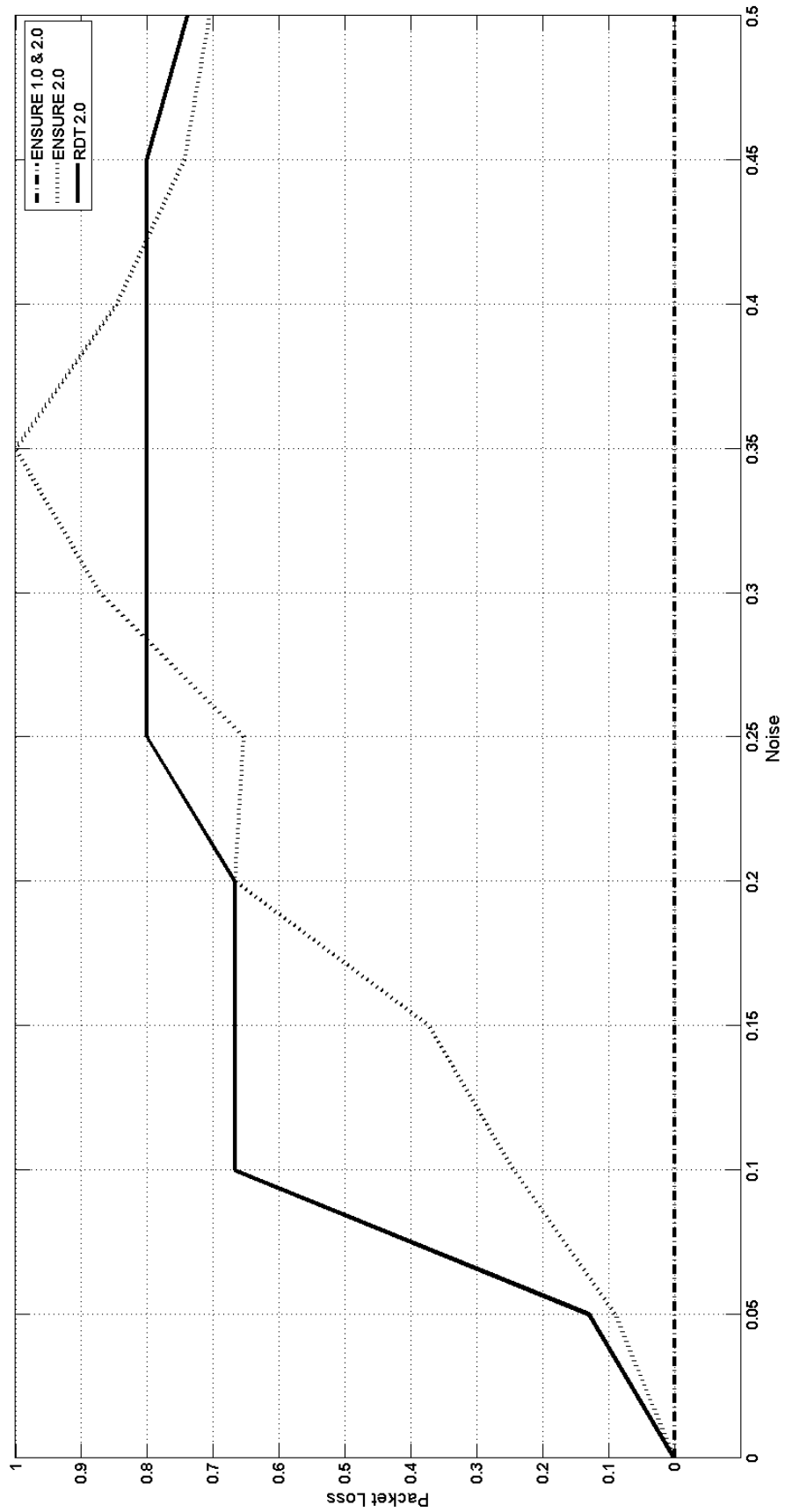


Figure 85. Protocol comparison of noise versus packet loss for three sensors

5.5.4 Reliability versus Packet Loss

Measuring reliability versus packet loss allows one to visualize the same results of the previous section in a different manner. Figures 83-85 provide further evidence that both ENSURE 1.0 and 3.0 provide complete reliability.

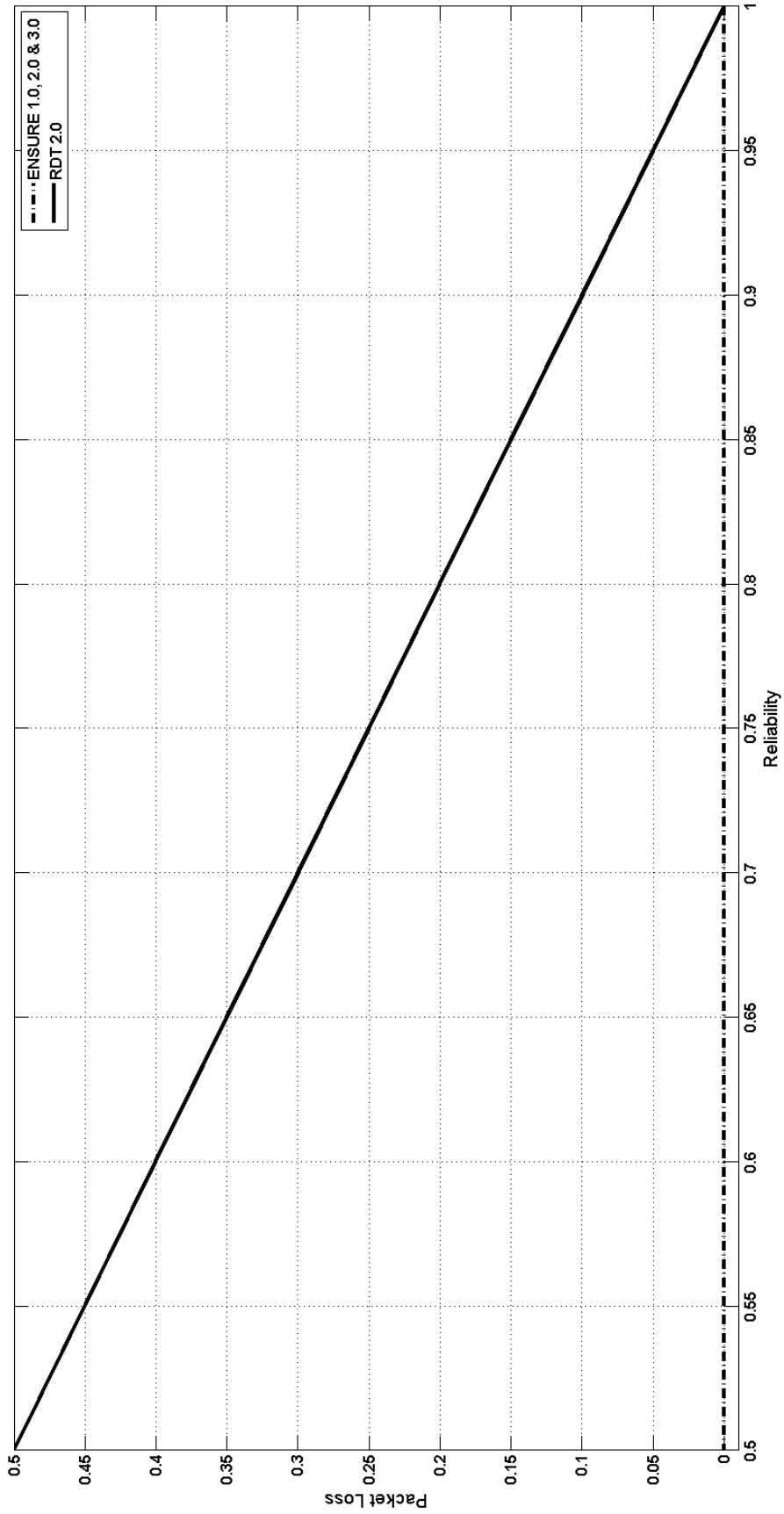


Figure 86. Protocol comparison of reliability versus packet loss for one sensor

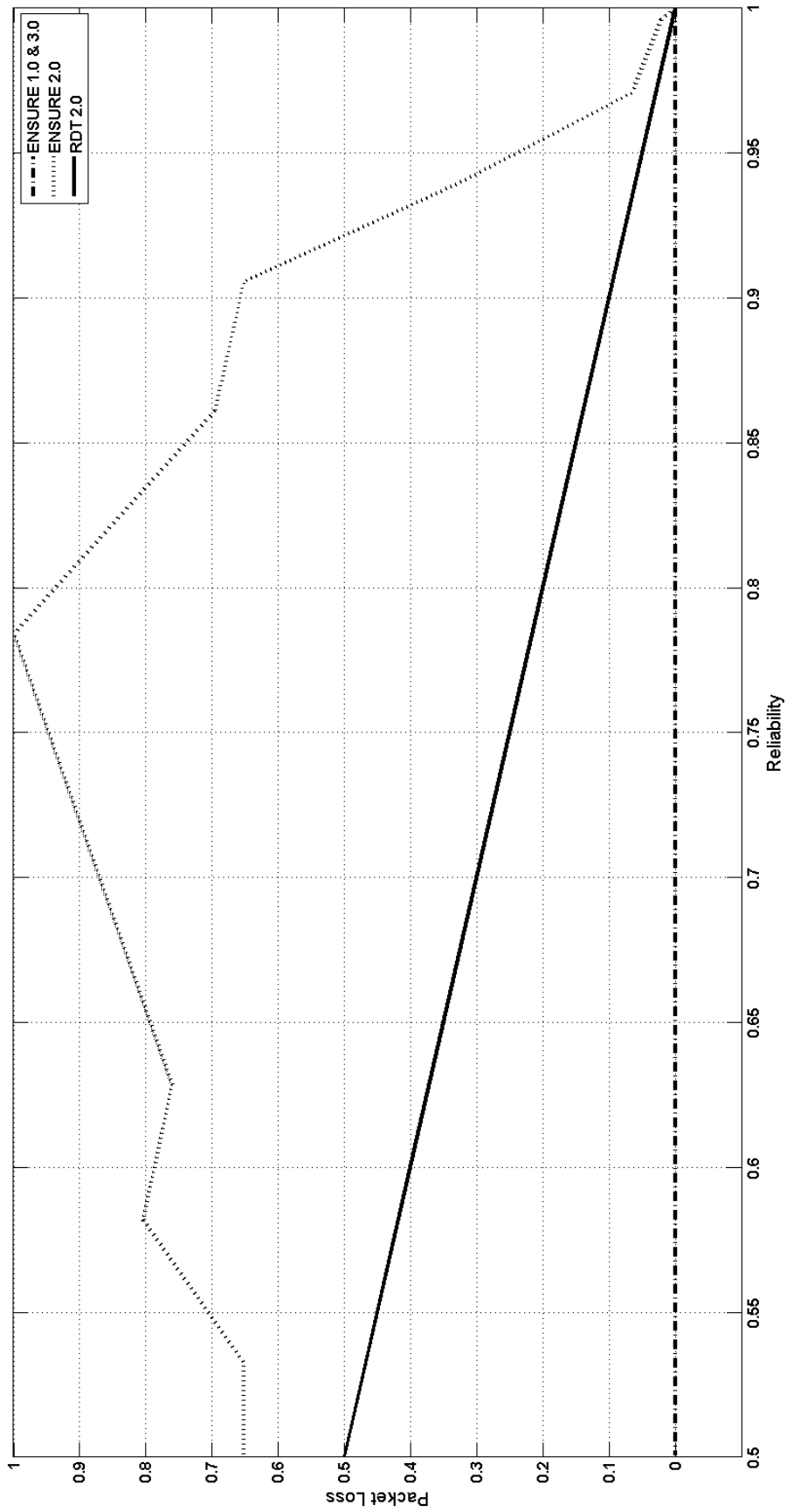


Figure 87. Protocol comparison of reliability versus packet loss for two sensors

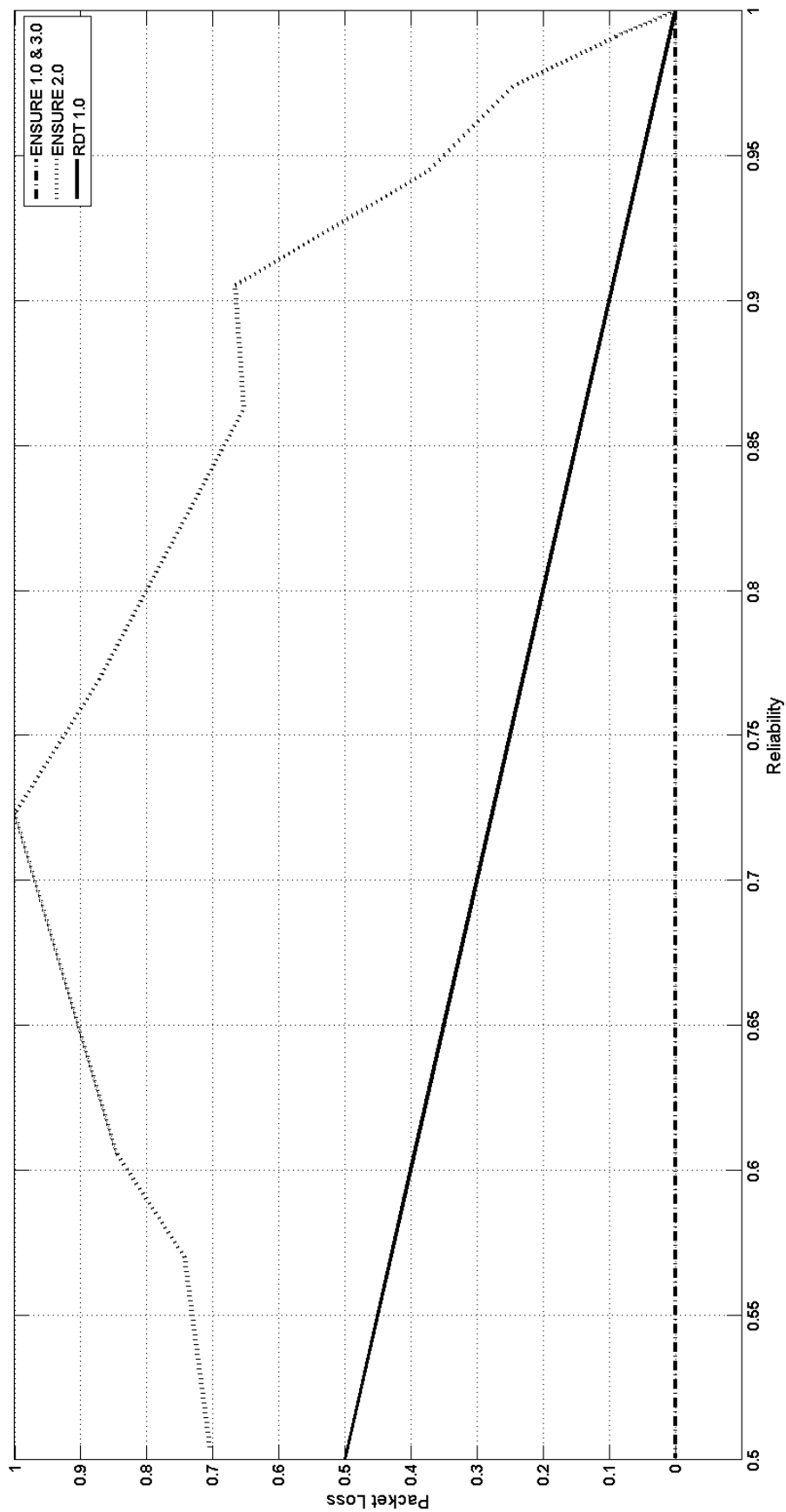


Figure 88. Protocol comparison of reliability versus packet loss for three sensors

Chapter VI

Conclusions and Future Research

The development of the series of ENSURE protocols successfully met the requirements of the petrochemical industry. Specifically, the simulation results for ENSURE 3.0 prove that 100% reliability can be achieved over a wireless channel even in extremely noisy settings and under time restrictions. Basic principles of a time sensitive network were explained by drawing a comparison to a military unit. This example demonstrated the need to adaptively make decisions based on current conditions. In real-time networking this principle holds true, and by incorporating current window size and packet loss ratio into the retransmission logic, channel conditions dictate the next action of the programmable logical controller.

This study established that a plant wide wireless implementation is not only feasible, but also can be controlled in three ways. An important take-away is that each PCL can run a different version of ENSURE, determined by the sensitivity class of the sensors. In other words, if simple logging of a gauge is necessary, perhaps ENSURE 1.0 is the right protocol for the job, for overall trends 2.0 would get the job done, and for 100% reliability delivered in a timely fashion ENSURE 3.0 has it covered.

The next phase of development is to utilize the CPSR Lab at The University of Texas at Tyler to implement the protocols on the SCADA equipment that replicates the functions of a petroleum plant. Also, future research can be done on the effects of using

different communication channels and adding more sensors to the network. The ultimate goal is to present a proven and practical ENSURE prototype to the industry. One integral part of the package will be to have built-in security. Research into the best way to lock down the communications can be done in parallel to the implementation stage. Further, an important question is: How security and reliability can be jointly met over noisy wireless petroleum plant channels? What level of overhead will be introduced to the system to achieve joint 100% reliability and 100% wireless communication security? These and other research questions represent possible future research directions of this work.

Works Cited

- [1] Sanja Sain, "Modelling and Characterization of Wireless Channels in Harsh Environments," Vasteras, June 2011.
- [2] Obaid ur Rehman and Natasa Zivic, "Iterative Bit Flip Type-II Hybrid-ARQ Scheme for Wireless Networks," in *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, vol. 27-29 April, Vienna, Austria, 2011, pp. 1-6.
- [3] Behrouz Maham and Merouane Debbah. (2011) Havard. [Online]. <http://adsabs.harvard.edu/abs/2012arXiv1202.2803M>
- [4] Hazilah Mad Kaidi and Muhammad Ibrahim, "Hybrid ARQ Type I Based on Convolution Code," *Jurnal Teknologi*, vol. 54, no. Jan. 2011, pp. 45-54, 2011.
- [5] Cong Shen and Micheal P. Fitz, "Hybrid ARQ in Multiple-Antenna Slow Fading Channels: Performance Limits and Optimal Linear Dispersion Code Design," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 5863 - 5883, Sept. 2011.
- [6] Chih-Chun Wang, Dimitrios Koutsonikolas, Y. Charlie Hu, and Ness Shroff, "FEC-based AP downlink transmission schemes for multiple flows: Combining the reliability and throughput enhancement of intra- and inter-flow coding," *Performance Evaluation*, vol. 68, no. 11, pp. 1116-1136, November 2011.
- [7] D. Gomez-Barquero, P. F. Gomez, D. Gozalvez, B. Sayadi, and L. Roullet, "A Novel Physical Layer Split FEC Scheme for Long Time Interleaving With Fast Zapping Support," *Broadcasting, IEEE Transactions on*, vol. 58, no. 2, pp. 269- 276, June 2012.
- [8] Kyunghoon Kwon, Hyun Ho Im, and Jun Heo, "An improved FEC system for next generation terrestrial 3D HDTV broadcasting," in *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, vol. Jan, Seoul, South Korea, 2012, pp. 327-328.
- [9] Liankuan Zhang and Deqin Xiao, "An Error-Aware Routing Protocol over Lossy Links in Wireless," *Procedia Engineering* 29, vol. 29, pp. 1492-1500, 2012.

- [10] M. Roshanzadeh and S Saqaeeeyan, "Error Detection & Correction in Wireless Sensor," *I. J. Computer Network and Information Security*, no. 2, pp. 29-35, 2012.
- [11] M. Mazzotti et al., "Analysis of Packet-Level Forward Error Correction," in *Vehicular Technology Conference (VTC Spring) 73rd*, vol. May, Cesena, Italy, 2011, pp. 1-5.
- [12] H.A. Ngo, S. Ahmed, Lie-Liang Yang, and L. Hanzo, "Erasure Insertion in Reed-Solomon Coded SFH M-ary FSK with Partial-Band Interference and Rayleigh Fading for Non-Coherent Cooperative Communications," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Southampton, UK , 2011, pp. 1 - 5.
- [13] S. Mehrotra, Jin Li, and Ying-Zong Huang, "Optimizing FEC Transmission Strategy for Minimizing Delay in Lossless Sequential Streaming ," *Multimedia, IEEE Transactions on*, vol. 13, no. 5, pp. 1066-1076, Oct. 2011.
- [14] J. Cervino, P. Rodriguez, I. Trajkovska, A. Mozo, and J. Salvachua, "Testing a Cloud Provider Network for Hybrid P2P and Cloud Streaming Architectures," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, Madrid, Spain, July 2011, pp. 356 - 363.
- [15] Floris van Nee and Pieter-Tjerk de Boer, "On the Benefit of Forward Error Correction," in *EUNICE Workshop*, vol. 6955, Enschede, The Netherlands, 2011, pp. 9 -20.
- [16] L. Xue, X. Guan, Z. Liu, and B. Yang, "TREE: Routing strategy with guarantee of QoS for industrial wireless sensor networks," *Int. J. Commun. Syst.*, pp. doi: 10.1002/dac.2376, May 2012.
- [17] Sohag Sarker, Farhana Enam, Md. Golam Rashed, and Shaikh Enayet Ullah, "Performance Analysis of Two-Layer Spreading scheme based FEC encoded MC-CDMA wireless communication system under implementation of various signal detection schemes," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 4, pp. 554 - 560, April 2012.
- [18] Cornelius Hellge, David Gómez-Barquero, Thomas Schierl, and Thomas Wiegand, "Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 551-561, June 2011.
- [19] Junni Zou, Hongkal Xiong, Chenglin Li, Rulfeng Zhang, and Zhihal He, "Lifetime and Distortion Optimization With Joint Source/Channel Rate Adaptation and Network Coding-Based Error Control in Wireless Video Sensor Networks," *IEEE Transactions on Vechicular Technology*, vol. 60, no. 3, pp. 1182-1194, March 2011.

- [20] M. Jonsson and K. Kunert, "Towards Reliable Wireless Industrial Communication With Real-Time Guarantees," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 4, pp. 429-442, Nov. 2009.
- [21] Xiaohui Liu, Hongwei Zhang, Qiao Xiang, Xin Che, and Xi Ju, "Taming uncertainties in real-time routing for wireless networked sensing and control," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, New York, NY, USA, 2012, pp. 75-84.
- [22] Muhammad R. Ahmed, Xu Huang, and Dharmendra Sharma, "A novel misbehavior evaluation with dempster-shafer theory in wireless sensor networks," in *MobiHoc*, Hilton Head, South Carolina, USA, ACM, pp. 259-260.
- [23] Md. Abdullah Al Mamun, Momotaz Begum, Sumaya Kazary, and Md. Rubel, "An Experimental Improvement Analysis of Loss Tolerant TCP (LT-TCP) For Wireless Network," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 10, pp. 77 - 80, 2011.
- [24] Xiaohui Liu, Hongwei Zhang, and Xiang Qiao. (2011) Wayne University. [Online]. <http://www.cs.wayne.edu/~hzhang/group/publications/MTA-RealWin.pdf>
- [25] K. Kunert, E. Uhlemann, and M. Jonsson, "Enhancing reliability in IEEE 802.11 based real-time networks through transport layer retransmissions," in *Industrial Embedded Systems (SIES), 2010 International Symposium on*, Halmstad, Sweden, July 2010, pp. 146 - 155.
- [26] K. Kunert, E. Uhlemann, and M. Jonsson, "Predictable real-time communications with improved reliability for IEEE 802.15.4 based industrial networks," in *Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on*, vol. May, Halmstad, Sweden, 2010, pp. 13 - 22.
- [27] Jesse Thomason, Kenji Yoshigoe, R. Lenin, James Bridges, and Srinu Ramaswamy, "Differentiated service strategies for ad-hoc wireless sensor networks in harsh communication environments," *Wireless Networks*, vol. 18, no. 5, pp. 551 - 564, July 2012.
- [28] R. Massin, C. Lamy-Bergot, C. J. Le Martret, and R. Fracchia, "OMNeT++-Based cross-layer simulator for content transmission over wireless ad hoc networks," *EURASIP J. Wirel. Commun. Netw*, vol. Apr, pp. 14:1--14:12, 2010.
- [29] Y. Isukapalli and B.D. Rao, "Packet Error Probability of a Transmit Beamforming System With Imperfect Feedback ," *Signal Processing, IEEE Transactions on*, vol. 58, no. 4, pp. 2298-2314, April 2010.

- [30] Xihua Dong, Xiaochen Li, and Dapeng Wu, "Analysis of Packet Error Probability in Delay Constrained Communication Over Fading Channels," in *Consumer Communications and Networking Conference*, Gainesville, FL, 2009, pp. 1 -5.
- [31] K. Fukawa, H. Suzuki, and Y. Tateishi, "Packet-Error-Rate Analysis Using Markov Models of the Signal-to-Interference Ratio for Mobile Packet Systems," in *Vehicular Technology, IEEE Transactions on*, vol. Vol. 61 Issue 6, Tokyo, Japan, 2012, pp. 2517 - 2530.
- [32] D.P. Andito, D.F.W. Yap, K.C. Lim, W.K. Yeo, and T.H. Oh, "Wireless sensor networks coverage: An error performance analysis," in *Communications (APCC), 2011 17th Asia-Pacific Conference on*, vol. Oct, Durian Tunggal, Malaysia, Oct 2011, pp. 836 - 839.
- [33] Binbin Chen, Ziling Zhou, Yuda Zhao, and Haifeng Yu, "Efficient error estimating coding: feasibility and applications," in *SIGCOMM '10*, vol. 14-4, New York, NY, USA, 2010, pp. 3-14.
- [34] Khairi Ashour Hamdi, "Packet-Error Probability Analysis for FH-CDMA Unslotted Packet Networks," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 151 - 154, 2003.
- [35] Hongyi Yao and T. Ho, "Error Estimating Codes with Constant Overhead: A Random Walk Approach," in *Communications (ICC), 2011 IEEE International Conference on*, Pasadena, CA, USA, June 2011, pp. 1 - 5.
- [36] Kyung Seung Ahn, R.W. Heath, and Heung Ki Baik, "Shannon Capacity and Symbol Error Rate of Space-Time Block Codes in MIMO Rayleigh Channels With Channel estimation Error," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 1, pp. 324 - 333, Jan. 2008.
- [37] Y.V. Zakharov, V.M. Baronkin, and J. Zhang, "Wireless Communications, IEEE Transactions on," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 2, pp. 617 - 621, Feb. 2009.
- [38] J. Ylioinas and M. Juntti, "Iterative Joint Detection, Decoding, and Channel Estimation in Turbo-Coded MIMO-OFDM ," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 4, pp. 1784 - 1796, 2009.
- [39] L. Dolecek et al., "Predicting error floors of structured LDPC codes: deterministic bounds and estimates ," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 6, pp. 908- 917, August 2009.

- [40] A. Bannour, M.L. Ammari, and R. Bouallegue, "Analysis of ASTC in a correlated Rayleigh fading channel with imperfect channel estimation," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2, Tunis, Tunisia, Feb. 2010, pp. 1300- 1303.
- [41] V. Mordachev and S. Loyka, "On node density - outage probability tradeoff in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 7, pp. 1120 - 1131, Sept 2009.
- [42] Ioanna Ioannou , Charalambos D. Charalambous , and Sergey Loyka. (2011, Feb) arXiv.org. [Online]. <http://arxiv.org/abs/1102.1103>
- [43] E.C.Y. Peh, Ying-Chang Liang, Yong Liang Guan, and Yonghong Zeng, "Power Control in Opportunistic Spectrum Access Cognitive Radio with Sensing Information at Transmitter," in *Communications (ICC), 2011 IEEE International Conference on*, vol. June, Singapore, Singapore, 2011, pp. 1 - 5.
- [44] Wei Wang and T Jost, "A Low-Cost Platform for Time-Variant Wireless Channel Measurements With Application to Positioning," vol. 61, no. 6, pp. 1597 - 1604, June 2012.
- [45] Dong Zhong and Li Zhongming , "The study of MIMO channel modeling and simulation based on 3GPP LTE," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, Xianning, China, 2012, pp. 3646 -3651.
- [46] Hu Xiaoya, Gao Chao, Wang Bingwen, and Wei Xiong, "Channel Modeling for Wireless Underground Sensor Networks," in *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, Wuhan, China, 2011, pp. 249 - 254.
- [47] S.S. Ikki and M.H. Ahmed, "Exact error probability and channel capacity of the best-relay cooperative-diversity networks," *IEEE Signal Processing Letter*, vol. 16, no. 12, pp. 1051 - 1054, Dec 2009.
- [48] K. Hijjeh, A. Mujahed, M. Jaber, S. Al-Hroub, and N. Dawadeh, "Matlab simulation for a MIMO channel modeling under bad urban and rural macro cell environments," in *Electromagnetics, Applications and Student Innovation (iWEM), 2011 IEEE International Workshop on*, Ramallah, Palestinian Authority, Aug. 2011, pp. 6 - 10.
- [49] Chia-Pang Yen and Peter J Voltz, "Indoor positioning based on statistical multipath channel modeling," *EURASIP Journal on Wireless Communications and Networking*, vol. 189, no. 1, 2011.

- [50] S. Jagadeesan and V. Parthasarathy, "Cross-Layer Design in Wireless Sensor Networks," in *Advances in Computer Science, Engineering & Applications*.: Springer Berlin / Heidelberg, 2012, pp. 283 - 295.
- [51] Zhi Sun, Pu Wang, Mehmet C. Vuran, Mznah A. Al-Rodhann, and Ian F. Akyildiz, "BorderSense: Border patrol through advanced wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 468 - 477, May 2011.
- [52] Pu Wang et al., "On network connectivity of wireless sensor networks for sandstorm monitoring," *Computer Networks*, vol. 55, no. 5, pp. 1150 - 1157, April 2011.
- [53] John A. Stankovic, Anthony D. Wood, and Tian He, "Realistic Applications for Wireless Sensor Networks," *Theoretical Aspects of Distributed Computing in Sensor Networks*, vol. 8, pp. 835 - 863, 2011.
- [54] S. De Vito et al., "Wireless Sensor Networks for Distributed Chemical Sensing: Addressing Power Consumption Limits With On-Board Intelligence," *Sensors Journal, IEEE*, vol. 11, no. 4, pp. 947 - 955, April 2011.
- [55] Luca Mottola and Gian Pietro Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 19:1--19:51, April 2011.
- [56] Kai Lin, Chin-Feng Lai, Xingang Liu, and Xin Guan, "Energy Efficiency Routing with Node Compromised Resistance in Wireless Sensor Networks," *Mob. Netw. Appl.*, vol. 17, no. 1, pp. 75 - 89, Feb. 2012.
- [57] Yean-Fu Wen, Tom A. Anderson, and David M. W. Powers. (2012, Feb) Wiley Online Library. [Online]. <http://dx.doi.org/10.1002/wcm.1249>
- [58] M. Sartipi and R. Fletcher, "Energy-Efficient Data Acquisition in Wireless Sensor Networks Using Compressed Sensing," in *Data Compression Conference (DCC)*, 2011, Chattanooga, TN, USA, 2011, pp. 223 - 232.

- [59] Kai Han, Liu Xiang, Jun Luo, and Yang Liu, "Minimum-energy connected coverage in wireless sensor networks with omni-directional and directional features," in *MobiHoc '12*, Hilton Head, South Carolina, USA, 2012, pp. 85-94.
- [60] Tsung-Chih Hsiao, Yu-Ting Liao, Jen-Yan Huang, Tzer-Shyong Chen, and Gwo-Boa Horng, "Secure Authentication Scheme for Supporting Healthcare in Wireless Sensor Networks," in *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, Taichung, Taiwan, March 2012, pp. 502-507.
- [61] Youngho Cho, Gang Qu, and Yuanming Wu, "Insider Threats against Trust Mechanism with Watchdog and Defending Approaches in Wireless Sensor Networks," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, vol. May, 2012, pp. 134 - 141.
- [62] Ashok Kumar Das, Pranay Sharma, Santanu Chatterjee, and Jamuna Kanta Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1646 - 1656, Sept. 2012.
- [63] L. Lightfoot, Y. Li, and J. Ren, "STaR: design and quantitative measurement of source-location privacy for wireless sensor networks," *Security Comm. Networks*, pp. doi: 10.1002/sec.527, 2012.
- [64] Guoliang Xing, Minming Li, Tian Wang, Weijia Jia, and Jun Huang, "Efficient Rendezvous Algorithms for Mobility-Enabled Wireless Sensor Networks," *IEEE Transaction on Mobile Computing*, vol. 11, no. 1, pp. 47 - 60, Jan 2012.
- [65] Fengjun Shang, Peng Liu, Zhe Geng, and Zhicheng Zhao, "QoS Routing Algorithm Research Based on the Node Occupation Degree Control for Wireless Sensor Networks," in *Recent Advances in Computer Science and Information Engineering*.: Springer Berlin Heidelberg, 2012, pp. 297 - 303.
- [66] Hossein Fotouhi, Marco Zuniga, Mário Alves, Anis Koubaa, and Pedro Marrón, "Smart-HOP: A Reliable Handoff Mechanism for Mobile Wireless Sensor Networks," in *Wireless Sensor Networks*.: Springer Berlin / Heidelberg, 2012, pp. 131 - 146.

- [67] A. Das, "A novel association rule mining mechanism in wireless sensor networks," in *Emerging Trends and Applications in Computer Science (NCETACS), 2012 3rd National Conference on*, March 2012, pp. 274 - 276.
- [68] Barry A. Cipra, *The Ubiquitous Reed-Solomon Codes.*: SIAM News, 1993, vol. 26-1.
- [69] H.A. Ngo, T.D. Nguyen, and L. Hanzo, "Amplify-forward and decode-forward cooperation relying on systematic Luby transform coded Hybrid Automatic-Repeat-reQuest," *IET Communications*, vol. 5, no. 8, pp. 1096-1106, 2011.
- [70] Zhongwei Si, Mattias Andersson, Ragnar Thobaben, and Mikael Skoglund, "Rate-Compatible LDPC Convolutional Codes for Capacity-Approaching Hybrid ARQ," in *IEEE Information Theory Workshop*, vol. Oct, Stockholm, Sweden, 2011, pp. 513-517.