Communication Protocol for Residential Electrical Demand Response in Home Devices

By

Monageng Kgwadi

A thesis submitted to
The Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute of Electrical and Computer Engineering

Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada

July, 2009

The undersigned recommend to the Faculty of Graduate Studies and Research

acceptance of the thesis

Communication Protocol for Residential Electrical Demand Response in Home Devices

Submitted by **Monageng Kgwadi,** M.A.Sc

In partial fulfillment of the requirements for
the degree of Master of Applied Science

Dr. Thomas Kunz
Thesis Supervisor

Dr. Howard Schwartz
Chair, Department of Systems and Computer Engineering

Carleton University

July, 2009

Abstract

Implementation of electrical demand response (DR) in home devices requires a robust and secure communication channel with which to deliver pricing and event messages. The RBDS network has been identified as a good candidate to deliver messages to DR enabled devices. Through simulations, we show that RBDS can be employed to effectively deliver DR messages. Effective targeting of devices is non-trivial in a broadcast network like RBDS. We propose an addressing scheme to efficiently group devices logically, by location, and individually over RBDS with minimal overhead. We also address the security concerns for DR messages delivered over RBDS by investigating three strong authentication protocols. Simulations show that devices up to 120km can receive authenticated messages with high probability but beyond that, messages experience losses due to larger messages introduced by signatures. ECDSA provides the strongest security but is more computationally expensive than BiBa and HORSE.

Dedications

Dedicated to my family for their support in all my endeavors. In memory of my parents Leteng and Keganeditse Kgwadi.

Acknowledgements

I would like to thank Professor Thomas Kunz, his leadership and overall contribution made this achievement possible. e-Radio's contribution with information on RBDS, field test data and support with off-air monitor software and equipment is immeasurable. Their overall contribution made the project possible. I would like to thank Pedro Villanueva for his valuable initial research on addressing.

Contents

List of Figures

List of Tables

List of Acronyms

AFI - Authority and Format Identifier

AM - Amplitide Modulation

ANSI - American National Standards Institute

AT - Address Type BCD - Binary Coded Decimal

BiBa - Bins and Balls

CC - Country Code

CCF - Current Carrier Frequency

CEC - California Energy Commission

CNLP - Connectionless Network Layer Protocol

CRC - Cyclical Redundancy Check

DN - Destination Network

DNP - Distributed Network Protocol

DR - Demand Response

DRI - Demand Response Infrastructure

DRRC - Demand Response Research Center

DSA - Digital Signature Algorithm

DSP - Domain Specific Part

DSP - Digital Signal Processing

DST - Digital Signature Transponder

ECC - Elliptic Curve Cryptography

ECC - Extended Country Code

ERP - Effective Radiated Power

FM - Frequency Modulation

FPGA - Field Programmable Gate Array

ECDSA - Elliptic curve Digital Signature Algorithm

GSM - Global System for Mobile communications

GSN - Global Subscriber Number

HORS - Hash to Obtain Random Subsets

HORSE - HORS Extended

HVAC - Heating Ventillation and Air Conditioning

IC - Identification Code

IDI - Initial Domain Identifier

IED - Intelligent Electronic Device

IEEE - Institute of Electrical and Electronics Engineers

ISO - International Organization for Standardization

ITSI - Individual TETRA Subscriber Identity

ITU - International Telecommunications Union

MAC - Media Access Control

NDC - National Destination Code

NS - Network Simulator

NSAP - Network Service Access Points

NSN - National Significant Number

NTC - National Territory Code

OPC - Operator Codes

OTcl - Object Tcl

PAC - Paging Area Code

PCD - Programmable Communicating Device

PCT - Programmable Communicating Thermostat

PI - Program Identification

PIN - Program Item Number

PIER - Public Interest Energy Research

PTY - Program Type

RBDS - Radio Broadcast Data System

RDS - Radio Data System

RF - Radio Frequency

RFID - Radio Frequency Identification

RSSI - Received Signal Strength

SEAL - Self Authenticating Values

SCADA - Supervisory Control and Data Acquisition

TC - Trunk Code

Tcl - Tool Command Language

TDMA - Time Division Multiple Access

TESLA - Time Efficient Stream Loss-tolerant Authentication

TETRA - Terrestrial Trunked Radio

TID - Target Identifier

TMC - Traffic Message Channel

TMCC - TETRA Mobile Country Code

UASA - Utility or Application Specific Addressing

UCA - Utility Communications Architecture

UID - Utility Identity

UMC - Utility Message Channel

VHF - Very High Frequency

XML - Extensible Mark-up Language

Chapter 1

Introduction

There are advanced efforts to employ demand response in residences to manage the peak demand of electricity. The use of demand response will engage consumers in the efficient operation of the power grid. Demand response programs allow customers to respond to prices, incentives or directives from utility companies. Customers are to be equipped with smart meters and programmable communicating devices (PCD) that support the use of demand response. PCDs are capable of communicating with the utility companies and perform energy conserving tasks for the customer, resulting in savings. Examples of PCDs are Programmable Communicating Thermostats (PCT's), in-home displays, smart appliances, water heaters and other high energy consuming devices.

1.1 Motivation

Successful implementation of demand response is dependent on consumer awareness, access to information, incentives, and enabling tools [19]. Information exchange is an important part of demand response programs. Pricing information and emergency signals have to be communicated to customers to facilitate demand response programs. Studies have been conducted to define the information exchange architecture for the demand response infrastructure [22]. The definition of information exchange mechanisms have, until now, been a high-level description of the functionality. There is no protocol adopted, to our knowledge, for use for demand response programs. Communications play a major role in the operation of demand response programs and other smart grid applications. A well-defined robust and flexible communication protocol is therefore necessary to allow devices to communicate to support demand response.

The Programmable Communicating Thermostat (PCT) system is an example of an enabling technology that would allow sustainable implementation of demand response for residential use. The PCT system is designed to allow demand response programs to be applied to the power consumed by heating and air-conditioning (HVAC) of residences. The system allows thermostats to receive pricing and emergency events broadcast over a wireless communication channel from the utility companies. The programmable thermostats operate in an automated manner. A customer programs the thermostat and the thermostat responds to event messages accordingly. There is no single standard communication protocol adopted for delivering the demand response messages to the thermostats. Therefore, there is a need to develop a communications protocol to deliver demand response messages to home owners. The communication protocol needs to be robust to allow easy interoperability between devices manufactured by different vendors.

Addressability of the devices wherether individual or collectively as groups needs to be established in the protocol. The deployment of demand response in residences will result in a large numbers of devices. Home owners may enroll in one or more demand response programs. Each group could be targeted by the utility company to receive group-specific messages. A customer may need to access their home device remotely via the network, hence each device needs to be addressable individually. Moreover, load management may require addressing based on device location when parts of the grid are experiencing difficulties. Careful planning for the addressing scheme is necessary to ensure flexible and efficient use of the addressing space.

Security is a major feature that needs to be accounted for at the conception of the protocol to ensure secure communications. Work has been done to determine possible threats that the PCT system is subject to [7]. The work provides a risk management approach to propose mitigation steps to address the security concerns. The work presented in [7] provides a high level description of mitigation steps. There is need to define a secure protocol with which the messages are communicated to the PCT's. The security solution adopted needs to be evaluated for performance impact on network resources. Security threats inherent to specific networks also require investigations to assess deployment feasibility. One such network is the Radio Broadcast Data System (RBDS), which is a wireless broadcast network.

1.2 Thesis Contribution

We propose a protocol to enable a generic application running on top of RBDS to deliver messages to home devices. Our solution interfaces the physical infrastructure to the application by offering services that are required by the application but not supported by the physical infrastructure. The solution is equivalent to the Strawman's reference design for demand response infrastructure enabling services layer [22]. Specifically we address issues concerning effective addressing of devices and secure communications. We evaluate the solution using simulations to determine the impact on network resources. The PCT system is an immediate beneficiary of such a protocol, hence our study uses the PCT system as a test case.

An initial study to characterize the RBDS network is carried out by simulations. The simulations are carried out on the Network Simulator (NS-2.30) tool. Modifications to the NS-2 tool are made to model the RBDS network accurately. Comparisons of the simulation results and physical data collected from field tests are used to verify and validate the model. The initial study results form a basis on which further development of the protocol is evaluated. Our initial study confirms that it is feasible to deliver messages over RBDS to home devices. Simulations show that messages can be delivered to receivers up to 140 km from the transmitter, making RBDS suitable for delivering messages for both urban and rural settings. Simulation results also show that the reception probability of messages is inversely proportional to message size. Message re-transmissions can be used to improve the reception probability and coverage area. Our initial studies to characterize the network are consistent with an independent study carried on physical field measurements in a consultation technical report [8] by Heschong Mahone Group Inc. for the Demand Response Research Center (DRRC).

Addressing schemes that allow efficient targeting of devices and groups of devices are also studied. Existing addressing schemes are investigated to obtain insights into the addressing scheme that could be employed for residential demand response applications. We propose an addressing scheme that meets the primary requirements for residential demand response applications. The proposed solutions allows for future extensions to be made as the electrical power grid evolves and new applications with similar requirements arise.

Communication security methods are investigated to identify security protocols that could be employed for a one-way broadcast communication channel. We identify three general purpose cryptographic protocols that could be used on the RBDS network from the literature. The three security protocols are modified for application on a one-way communication channel. The impact on network resources of the security protocols is studied by using simulations. Comparisons are then drawn between the three candidate protocols. The three candidate security protocols offer strong source authentication. ECDSA provides the strongest authentication at the expense of high computational complexity. HORSE and BiBa have low complexity but require device bootstrapping and have larger public keys. Consistent with our initial study, the increase in message size due to digital signatures reduces message reception probability, and the coverage area. Simulations show that ECDSA and HORSE outperform BiBa in terms of message reception probability due to the inefficient use of public keys in BiBa. Simulations also show that receivers up to 120 km can be reached with high reception probability.

The results of the work have featured in regular reports presented to industrial partners and consulting interactions. A technical report on security has been published through the Department of Systems and Computer Engineering (SCE-09-06) presenting our findings. The source code used for simulations has also been provided to partners to aid them in further studies and is available upon request. The work has also been showcased at the Ontario Centres of Excellence discovery day in May 2009.

1.3 Thesis Organization

A background on demand response programs and enabling technologies is presented in Chapter 2. Chapter 3 presents a brief background of the Radio Broadcast Data System. Chapter 4 presents the initial study of the RBDS network. It explains the simulation models used to represent the network. Results of the initial study are also presented in Chapter 4. Our study on addressing of devices is presented in Chapter 5, we then present our addressing proposal Section 5.3. Chapter 6 presents the study on the security of the communication protocol. We discuss the threat model as it applies to the RBDS network and mitigation steps in Section 6.2. We review the literature for applicable cryptographic methods and shortlist three candidate authentication protocols in Section 6.3. We identify and present three authentication protocols and show how they can be used over RBDS in Section 6.4. The network impact of the proposed protocols is investigated and presented in Section 6.5. Conclusions and future work are finally presented in Chapter 7.

Chapter 2

Smart Grid and Demand Response

2.1 Smart Grid

Recent developments in the power industry have led to the proposal and adoption of smart grids in North America and Europe. A smart grid is defined as an integration of electrical power infrastructure and information systems to improve performance of the electrical power grid [30]. It uses innovative products and services such as automation, control, sensors, and communications to effectively and economically offer high reliability services by engaging both power generators and consumers. With the use of smart grids, it is possible to integrate different power generation technologies of varying sizes into the grid. The availability of information and choices allows consumers to play an active role in the operation of the power grid. Ultimately the smart grid is envisioned to minimize the environmental impact of the electrical power grid and enhance performance.

Figure 2.1 shows components of a smart grid. Applications communicate with consumers, control sensors, and power resource systems over a communication infrastructure. Examples of applications running in a smart grid are supervisory control and data acquisition (SCADA), demand response, forecasting of power markets, etc.

As shown in Figure 2.1, different power generation technologies can be integrated into the power grid. Communications play a vital role to bring together the power infrastructure and the different applications required to operate and manage the system.

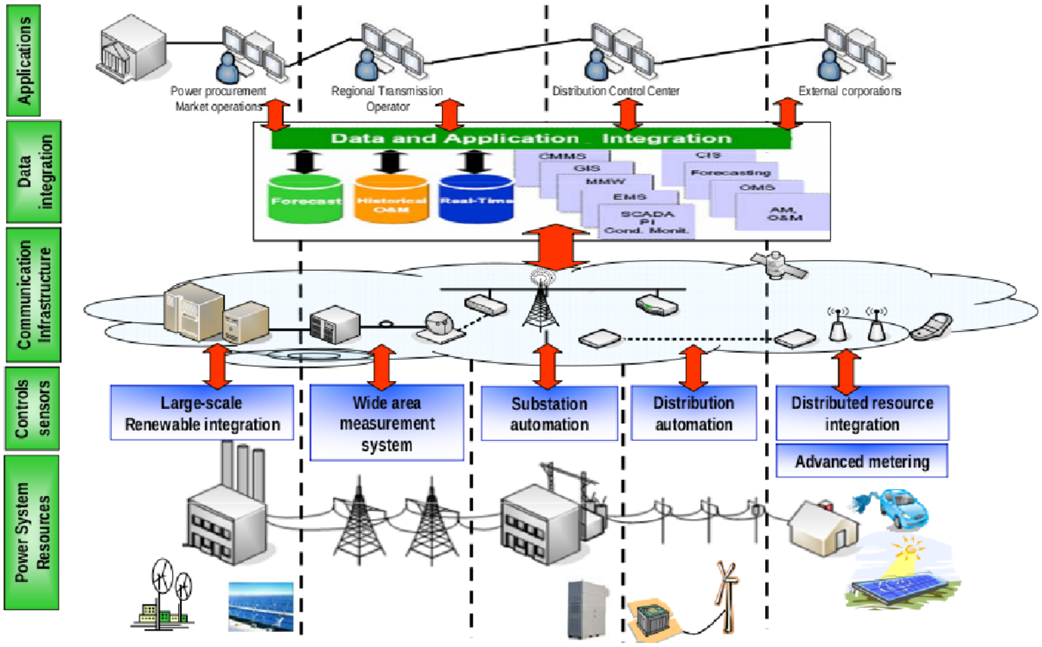

Applications
Data integration
Communication Infrastructure
Controls sensors
Power System Resources
Power procurement Market operations
Regional Transmission Operator
Distribution Control Center
External corporations
Data and Application Integration
Forecast
Historical O&M
Real-Time
CMMS
GIS
MMW
EMS
SCADA PI Cond. Monit.
CIS
Forecasting
OMS
AM, O&M
Large-scale Renewable integration
Wide area measurement system
Substation automation
Distribution automation
Distributed resource integration
Advanced metering

Figure 2.1: Architecture of a Smart Grid [9]

Large scale electrical generation needs to be throttled to match the demand in real-time because electricity cannot be stored efficiently without significant losses [19]. Electrical utility companies are tasked with matching the supply with the changing electricity demand to avoid mismatches between generation and load. In order to offer high availability of services, power companies must have the capacity to handle maximum peak load regardless of how often it occurs. Maximum peak load generally occurs infrequently and lasts for short periods. In California, 10% of the power generation capacity is only used for approximately 1% of the time to generate 0.1% of the total energy consumed by the State [19]. Figure 2.2 shows the electrical load for the state of California for 2005 and 2006. As shown in the figure, demand in excess of 54 536 MW occurred for less than 1% of the time in 2006. The power generation plants that are meant to meet maximum peak demand stay idle for the majority of the time. The costs of these resources are recovered in the short period that they are active, resulting in high prices for consumers. Moreover, periods with excessive loads result in inefficient electricity generation [19]. During peak-demand periods, power companies activate generation plants in decreasing order of efficiency.

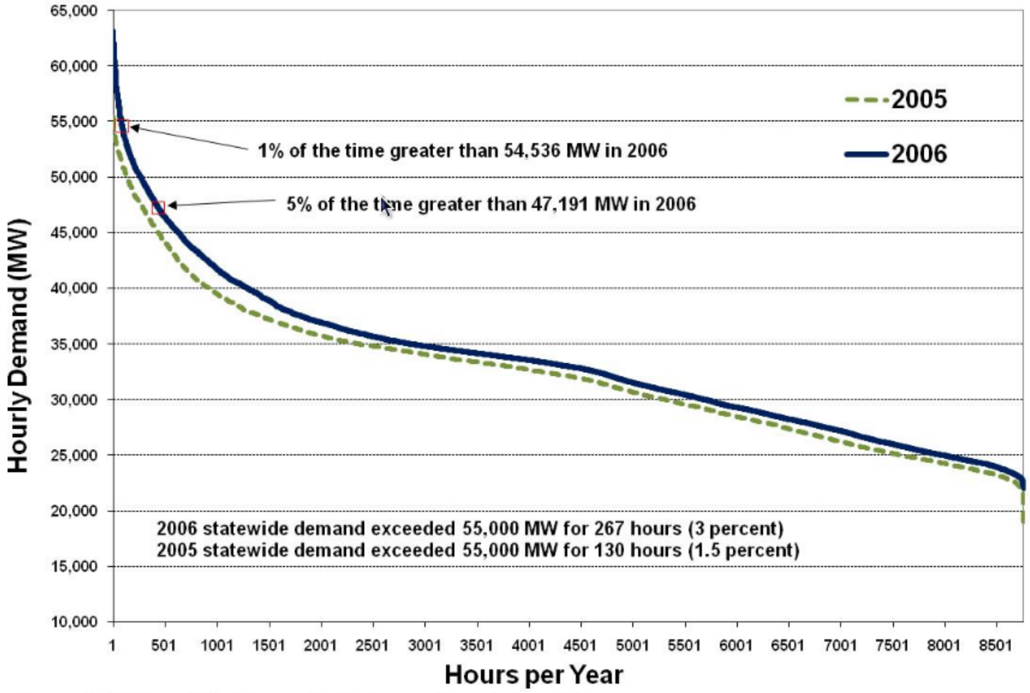


Figure 2.2: Electrical Load Curve for Carlifornia (2005-2006) [19]

2.2 Demand Response Programs

Demand response is an active response in consumption of electricity by customers to prices, incentives or directives from electricity utility companies [6]. Demand response programs enable electric power consumers to reduce consumption during a certain period, or shift consumption to a different period. The change in consumption could be in response to pricing signals, financial incentives, environmental conditions, or emergency signals [19]. Demand response programs present grid operators another alternative besides additional generation, purchase, or load shedding, to address excess peak demand and grid instabilities. Load management through demand response programs is achieved by giving customers incentives for reducing their consumption during peak demand periods, or by shifting their consumption to off-peak periods. During emergencies, when grids experience instabilities, demand response can be used to reduce the load by the use of emergency signals. The shift of consumption or reduced consumption by customers during peak demand periods reduce excessive loads on the grids which would otherwise require additional power generation or purchase.

Demand response is projected to save 202TWh of energy annually in Europe by 2020 [14]. Moreover, the amount of CO_2 emissions can be reduced by 100 million tons through the use of demand response annually. Further €50 billion savings in power generation capacity to meet peak demand can be achieved with the use of demand response. Customers are expected to save up to €25 billion in electricity bills annually with the use of demand respose programs.

Demand response programs are enabled by the evolution of technologies in metering and information [32]. Meters have evolved from recording gross power consumption to smart meters that record the amount of consumed power in a certain period of time. These new meters can send out the readings via a wireless link to a meter reader when queried. The use of such advanced metering equipment makes the realization of demand response programs feasible. There are advanced developments on the implementation of demand response programs and the topic is an active research field. The University of Berkeley, together with the Government of California, has done a lot of work to realize the use of such programs. The California Energy Commission; Public Interest Energy Research (CEC-PIER), together with the California Programmable Communicating Thermostats Collaboration developed a reference design for Programmable Communicating Thermostats (PCT) to be used in homes. The PCT system introduces demand response for air conditioning systems in homes. It is envisioned that customers would forgo a few degrees centigrades of temperature settings during emergency situations to avoid blackouts. Customers could also respond to dynamic electricity prices and incentives by moderate use of air conditioning during periods of high prices to reduce their power bills. The use of demand response programs on the HVAC utility is at an advanced stage with the reference design now sent to vendors to start designing and fabricating the programmable communicating thermostats.

In addition to the above use of messages to enable users to respond to dynamic energy prices in residences, direct load control does not use dynamic energy prices. Instead, consumers are provided special rates or other incentives for allowing the utility to control load (typically air conditioning) for a number of days per year. Also, the original PCT system concept has since been expanded to include a range of Programmable Communicating Devices (PCDs), including PCTs, in-home displays, smart appliances and control switches for air conditioning, water heaters and other high energy consuming devices. In future extensions, plug-in hybrid vehicles could also be included. However, in all these extensions of the original idea, the overall concept is the same: utilities send messages to devices to inform users and to potentially directly control the load. In the remainder of the thesis, we will use the PCT system as the example PCD system, as this work was started with respect to the requirements identified for the PCT system. But the key insights and solutions apply equally well to a more general definition of a receiver device, be it a thermostat, in-home displays, or smart appliances.

2.3 The Programmable Communicating Thermostat System

The Programmable Communicating Thermostat (PCT) system is envisioned to eliminate or reduce rotating outages as a measure of dealing with power emergencies [23]. The PCT system will instead offer load reduction by temporary reduction in air-conditioning services. The system allows customers to automate their responses to dynamic pricing of electricity. Pricing information for electricity is communicated to customers to allow them to keep their electricity bills low. The customers program the PCTs to respond to event messages automatically. Incentive programs offered to customers for participating in load reduction and demand response programs are also proposed through the PCT system. The PCT system would also support system reliability by compulsory load shedding of HVAC services when there are grid instabilities [23]. The load shedding is a last resort means of averting power outages and customers have no way of overriding such directives from utility companies.

Partial outages by the elimination of a single discretionary service such as air-conditioning are preferred to rotating outages. Rotating outages takes out all of the services, hence partial outages are more economically efficient [23]. The PCT targets the HVAC services for partial outages because it is considered discretionary. In the event of extreme emergencies, it is believed that reducing air-conditioning by a few degrees would not have a significant impact on customers. Partial outages of air-conditioning services can account for both small and large amounts of energy savings(up to 30% in California during peak periods) and anywhere in between.

The reference design for PCTs outlines the implementation of the HVAC, Human-Machine, Communications, and Expansion Interfaces in terms of hardware and software characteristics. The design has gone through numerous revisions and refinements to allow the specifications to be as detailed as possible, yet at the same time not hindering future developments. Hence the revision is not technology specific to be able to be upward compatible with changing technologies.

2.3.1 PCT Communications

The communications architecture for the PCT system is based on the Strawman demand response infrastructure consulting solution for the CEC [22]. The solution describes a layered architecture with three major layers: the physical infrastructure, enabling services, and application layer. Each of the layers has functional, control and trust features. Figure 6.6 shows the layers of the Strawman reference design for demand response information exchange.

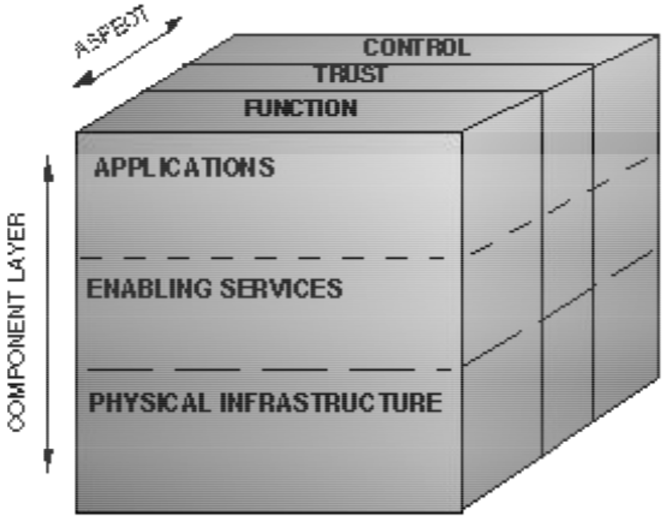


Figure 2.3: Architecture of the Demand Response Infrastructure [22]

The physical infrastructure consists of a combination of components that make up a communications system. The reference design for PCTs mandates a compulsory wide area communications interface using the Radio Broadcast Data System (RBDS) or paging system [23]. An optional network interface fitted over the expansion interface overwrites the use of the default broadcast interface. The default non-removable wide area communications interface is one way communication to facilitate the receipt of demand response (DR) messages from the utility companies. Disseminating event messages for the PCT system makes the use of broadcasting techniques attractive. The optimal medium of transmission of messages to PCTs is an open research topic and hence the expansion interface allows many communications technologies to be used with the PCTs. For the default communication interface, the RBDS is a good candidate because of the physical characteristics of FM transmission. Future developments will see two-way communications between the utility companies and the PCTs.

The enabling services layer provides a common platform for different applications to use the physical layer. The enabling services layer performs system-level functions to allow new services and applications to inter-operate with the existing infrastructure and applications. The enabling services provide an application programmer interface (API) to aid application development. The applications depend on the underlying enabling services and physical infrastructure to perform some duties for the users. In the case of the PCT system, the application handles the operation of the HVAC equipment in a customer's home. The enabling services aid the operation of demand response programs by ensuring delivery of information to the application. Price events notify the customers of a change in the price of the utility. The PCTs respond to price events by the pre-programmed instructions that the user can overwrite manually when necessary through the human-machine interface. The emergency events notify the customers of an emergency situation that requires load reduction due to grid reliability issues. On receipt of an emergency event message, the PCTs change the operation settings accordingly. The customers are not allowed to overwrite adjustments initiated by an emergency event.

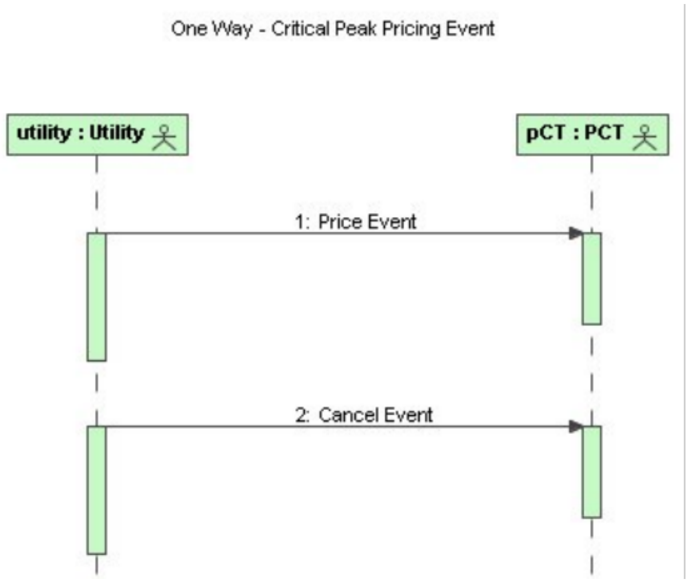


Figure 2.4: One-Way Pricing Event [23]

Figure 2.4 and Figure 2.5, respectively, show how price and emergency events are communicated to the PCT using a one-way communication channel. In a one-way pricing event, the utility company announces a price for electricity. The utility can cancel the price at a later time when the price of electricity changes. The emergency events work in the same way, and Figure 2.5 shows how a 2 stage emergency event is

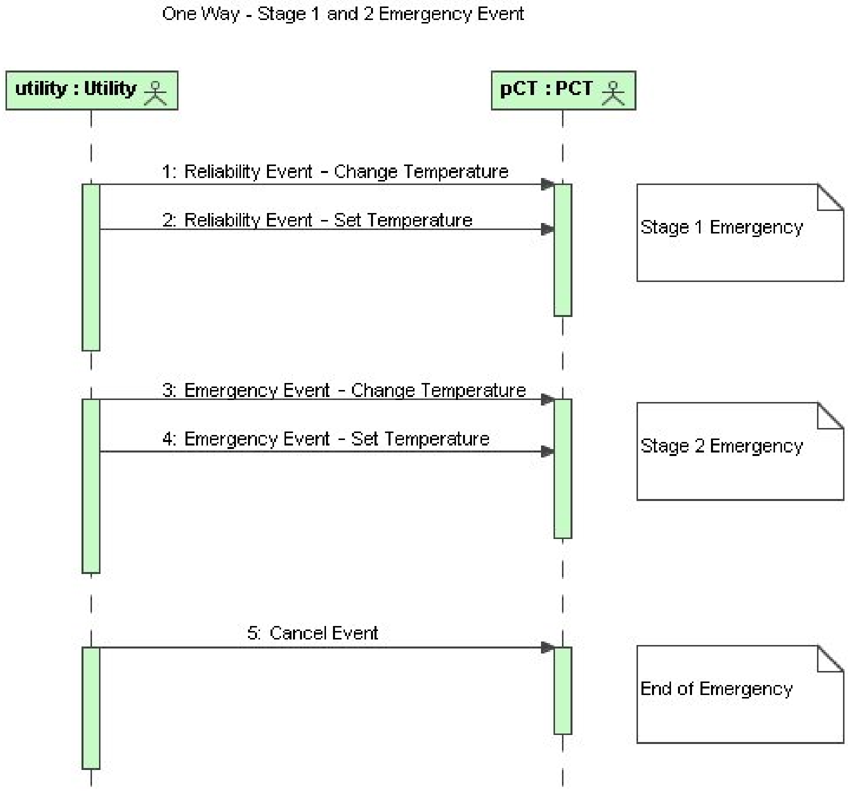

One Way - Stage 1 and 2 Emergency Event
utility : Utility
pCT : PCT
1: Reliability Event - Change Temperature
2: Reliability Event - Set Temperature
Stage 1 Emergency
3: Emergency Event - Change Temperature
4: Emergency Event - Set Temperature
Stage 2 Emergency
5: Cancel Event
End of Emergency

Figure 2.5: One-Way 2 Stage Emergency Event [23]

sent to devices in a one-way communication channel.

2.3.2 PCT Messages

The reference design also specifies that the representation of the messages sent to the PCTs should be in Extensible Mark-up Language (XML). The proposed fields, sizes, and data types of the messages are also defined in the specifications without explicitly emphasizing on the exact format of the message. XML has been shown to be orders of magnitude larger than other formats. The impact of the messages on the network becomes an issue of concern when using XML over potentially bandwidth constrained links like the RBDS. Compression techniques for XML messages come to the fore in order to investigate the most efficient way of sending the messages. There are several existing compression techniques for XML data. The techniques offer different compression rates and complexities based on the sizes of the compressed messages.

Chapter 3

Enabling Technologies

Successful implementation of demand response requires a communication channel(s) to enable transmission of messages. e-Radio Inc. has a Utility Message Channel (RDS UMC) that offers utility companies and demand response providers a way of sending messages to thermostats, appliance controllers and in-home display units [25]. Their service uses the Radio Broadcast Data System (RBDS) to relay the messages to the appliances. RBDS offers enhanced audio and data delivery functionalities on the FM channel. The existing infrastructure makes the use of RBDS financially attractive since there will be no need for a new network deployment. RBDS is particularly attractive for a state-wide network since the network already covers most of the homes in North America, even in remote areas. FM transmission offers good building penetration and is highly available. e-Radio USA already offers traffic information to drivers through the Traffic Message Channel (RDS TMC) [25]. The application offers relevant traffic information to a driver based on the driver's location. Other messages delivered to drivers on their car display units include news, weather, sports results, spot advertising, song titles, and artists.

The Utility Message Channel network as defined by e-Radio offers services to major utilities, not only specific to electrical power and demand response providers. The e-Radio RDS-UMC network is depicted in Figure 3.1. Utility companies send messages to the e-Radio Operations Centre. Then using the RBDS network, e-Radio transmits the messages to the PCTs via a radio broadcast service from an FM radio station.

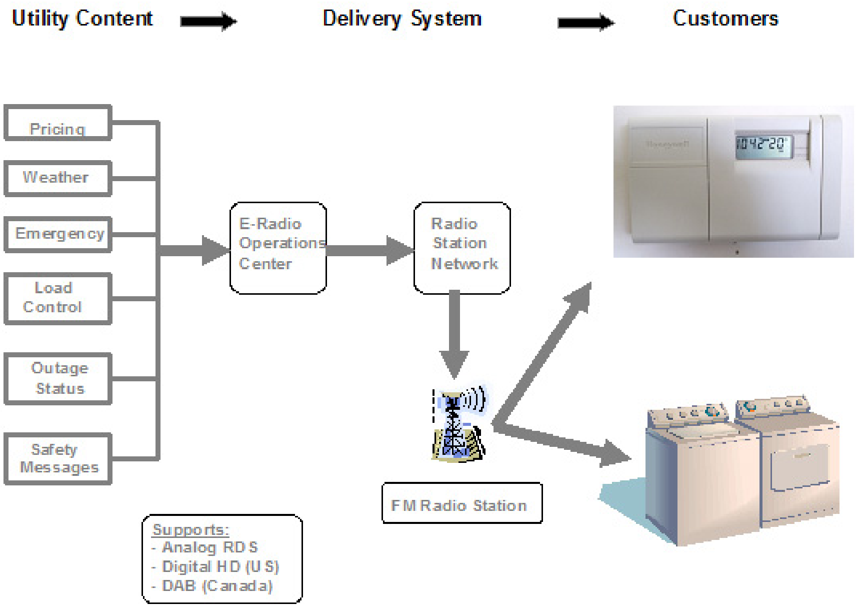


Figure 3.1: e-Radio Utility Message Channel [25]

3.1 The Radio Broadcast Data System

The Radio Broadcast Data System (RBDS) is used for the transmission of small packets over the FM channel. RBDS is a North American radio broadcast standard equivalent to a European standard, the Radio Data System (RDS). The basic functionalities of RBDS and RDS are identical [39]. RBDS completely contains RDS in its entirety with additional features added to RBDS. In this document the terms RBDS and RDS are used interchangeably.

RBDS is designed for VHF/FM transmitters in the frequency range of 87.5MHz to 108.0MHz with stereophonic (pilot tone) or monophonic sound broadcasts [2]. The system uses a subcarrier locked in phase or quadrature to the third harmonic of the 19KHz pilot tone for stereo broadcasts and a 57KHz subcarrier for stereophonic broadcasts [2]. RBDS transmits data bits using amplitude modulation on the subcarrier with a datarate of 1187.5 bits/sec [2, 28].

RBDS sends data in groups of 104 bits in size. An RBDS group is made of four 26-bit blocks, each block is made up of 16 data bits and 10 error correction bits. The error correction employed by RBDS is cyclical redundancy check (CRC). The transmission of bits is synchronous and there are no gaps between successive groups or blocks. Figure 3.2 shows the scheduling of an RBDS group for sending.

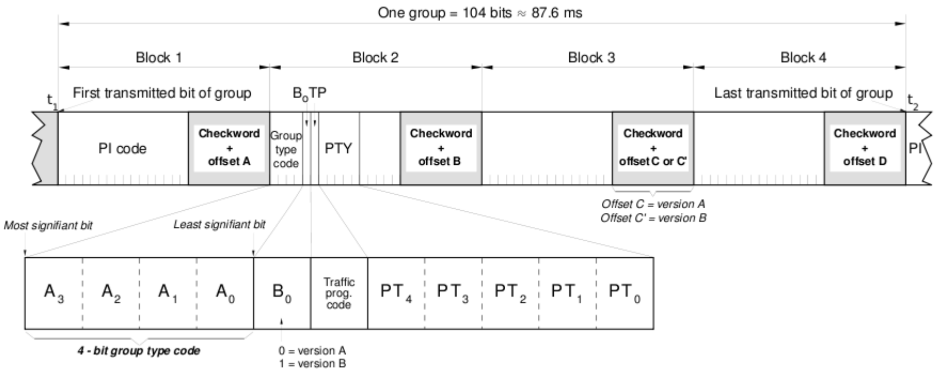


Figure 3.2: RBDS Group and Blocks [2]

The first block (Block 1) carries the Program Identification (PI) code, which uniquely identifies a radio station [28]. This information is repeated 11.5 times every second. Block 2 includes the Program Type (PTY) code, Group Type code, and

Traffic Program Identification code. The Group Type code in Block 2 defines what data is carried in Blocks 3 and 4.

3.2 RBDS Messages and PCT Messages

There are thirty-two RBDS group types which carry different types of data. The data carried by an RBDS group ranges from application-specific data to general data. See Appendix B for a list of group types and descriptions of the data carried by each group type. Pilot projects for the PCT system use Group 11A to transmit messages over the RBDS network. The group types 11A and 11B are classified under the RBDS standard to carry open data. Figure 3.3 shows the format of group type 11A.

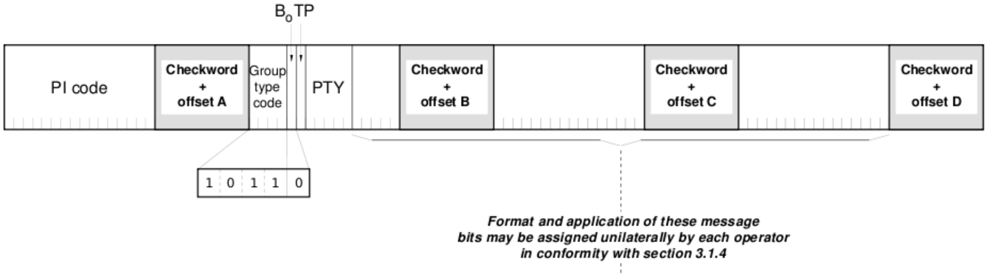

B0 TP
PI code
Checkword + offset A
Group type code
PTY
Checkword + offset B
Checkword + offset C
Checkword + offset D
1 0 1 1 0
Format and application of these message bits may be assigned unilaterally by each operator in conformity with section 3.1.4

Figure 3.3: Group Type 11A [2]

Block 1 of an RBDS group type 11A is used to convey the Program Identification (PI) code. Block 2 conveys the type of data in Blocks 3 and 4 and in the case of PCT messages, signaling bits. The signaling bits in Block 2 convey fragmentation information that allows messages to be reconstructed from fragments. The last five data bits in Block 2 can be used by each operator to suit their needs as shown in Figure 3.3 and stated in [2]. Since data can only be placed in Blocks 3 and 4, messages that are larger than 4 bytes are fragmented and transmitted in more than one RBDS group. Bit 4 of Block 2 specifies that the data carried in Blocks 3 and 4 is the first packet of a fragmented message. Bit 3 specifies a last packet of the received message is contained in Blocks 3 and 4. A one packet message will have both Bit 4 and Bit 3 set to 1. Bits 0-2 carry a 3-bit message ID which increments with each unique message sent. The message ID wraps around after every 8 messages.

RBDS sends PCT messages in two modes: the normal mode and the real-time mode [37]. In the normal mode of operation, messages are sent in a synchronized manner at predefined times. In the real-time mode the messages are sent as soon as they are received from the application and experience minimal delay in the outgoing queue.

RBDS Normal Mode

In normal mode, [37] defines a minimum time between successive unique messages, NTX_new. The normal mode enables receivers to go into power save mode and wake up at the beginning of each NTX_new period. If no new message is received within DRX_continous seconds at the start of a NTX_new period, the receiver can go into battery save mode (sleep). Each message is repeated NTX_repeat times to increase chances of reception. Figure 3.4 shows how the transmission of two PCT messages in RBDS is scheduled as described in [37]. Messages larger than 4 bytes are fragmented into multiple packets and transmitted NTX_repeat (n) times. During NTX_new, a message is transmitted NTX_repeat (n) times. If the message is fragmented into multiple packets, all the packets are transmitted NTX_repeat times in one NTX_new time frame. The first packet is transmitted NTX_repeat times, followed by the second packet and so on until all fragments are transmitted [37]. At most two PCT messages can be transmitted simultaneously as shown in Figure 3.4.

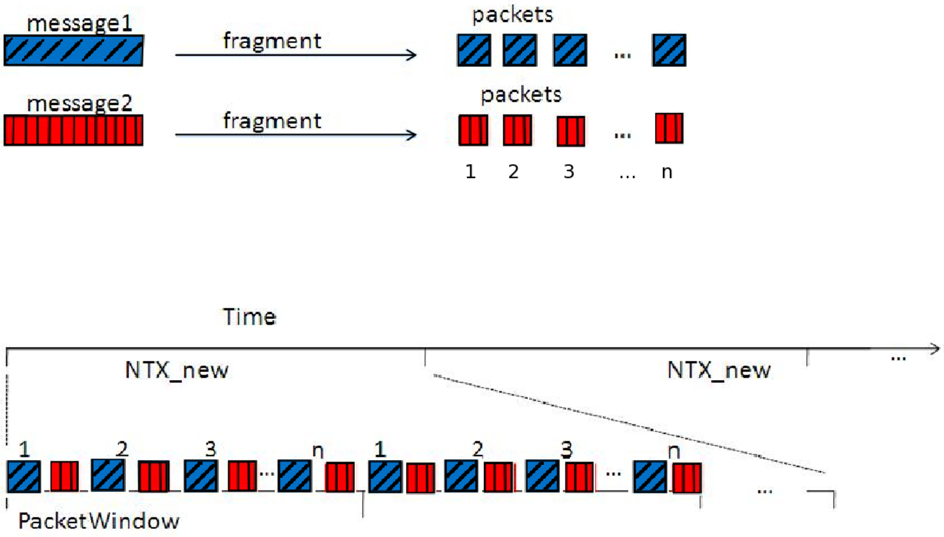


Figure 3.4: RBDS Scheduling of PCT Messages

In normal mode, when a new message arrives, it is placed in the outgoing buffer. At the end of each NTX_new time frame, the outgoing buffer is checked for messages that arrived, and at most two messages are dequeued and sent out as shown in Figure 3.4.

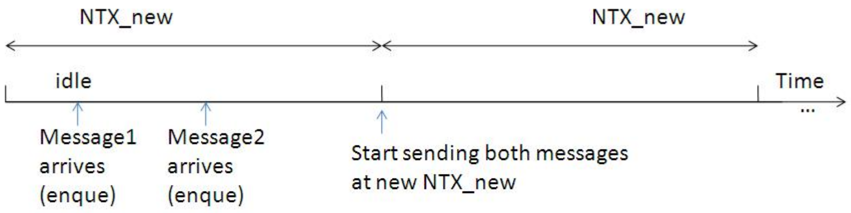


Figure 3.5: RBDS Normal Mode Operation

Figure 3.5 illustrates the dynamics of the RBDS in normal mode operation. The first message is received from upper layers and placed in the buffer until the next NTX_new time frame starts. The second message arrives before the start of the next NTX_new time frame and is also placed in the outgoing buffer. At the beginning of the next NTX_new time frame, the first packets of the two messages are transmitted as shown by Figure 3.4.

RBDS Real-Time Mode

In real-time mode, messages are sent immediately upon reception from the application. If there are currently two messages being transmitted and a new message comes, the older of the two messages is preempted by the new one as described in [37].

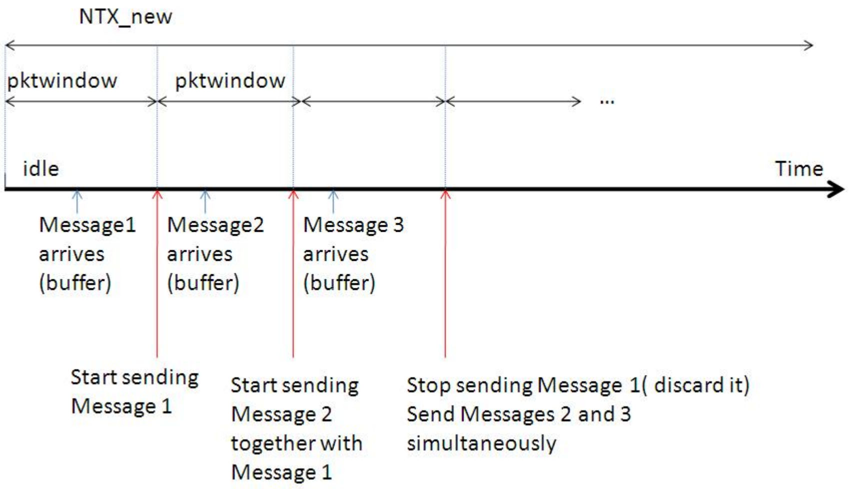

NTX_new
pktwindow
pktwindow
...
idle
Time
Message1 arrives (buffer)
Message2 arrives (buffer)
Message 3 arrives (buffer)
Start sending Message 1
Start sending Message 2 together with Message 1
Stop sending Message 1(discard it) Send Messages 2 and 3 simultaneously

Figure 3.6: RBDS Real-Time Mode Operation

Figure 3.6 shows the dynamics of the real-time mode. Initially the transmitter is idle, a message (Message1) is then received from the application for transmission. At the start of the next PacketWindow, the first packet of Message1 gets transmitted. During the transmission of Message1, Message2 arrives and gets buffered until the current PacketWindow elapses. At the beginning of the next PacketWindow, the second packet of Message1 continues to be transmitted (assuming it has more than one fragment). In the same PacketWindow, the first packet of Message2 gets transmitted. Message3 arrives while both messages are being transmitted and placed in the buffer until the current PacketWindow elapses. At the next PacketWindow, Message1 is pre-empted by Message3 and is discarded. The first packet of Message3 and the second packet of Message2 are then sent during the PacketWindow.

At the receiver end, the messages are re-constructed from the fragments. The reception of messages uses sequence numbers to identify the packets and reassemble a message. Each first fragment of a message is identified by a FirstPacket bit (bit 4) set in the signaling field in Block 2. The last packet is also identified by a LastPacket bit (bit 3) in Block 2. The intermediate messages are identified by the MessageID and sequence numbers. To distinguish between intermediate packets of one message, the receiver uses sequence numbers. The fragments are numbered 0 through 15 and wrap around if the message has more than 16 fragments. Every time a packet with a new MessageID and FirstPacket bit set is received, it is placed in one of the two incoming message buffers. If both buffers are occupied, the buffer containing the older message is cleared and used to receive the new message. When a packet is received with the LastPacket bit set, then the incoming buffer is checked for the number of received fragments. If fewer fragments are received than the number of elapsed PacketWindow time frames, the message can not be reconstructed and is discarded.

Chapter 4

Network Modeling and Model Calibration

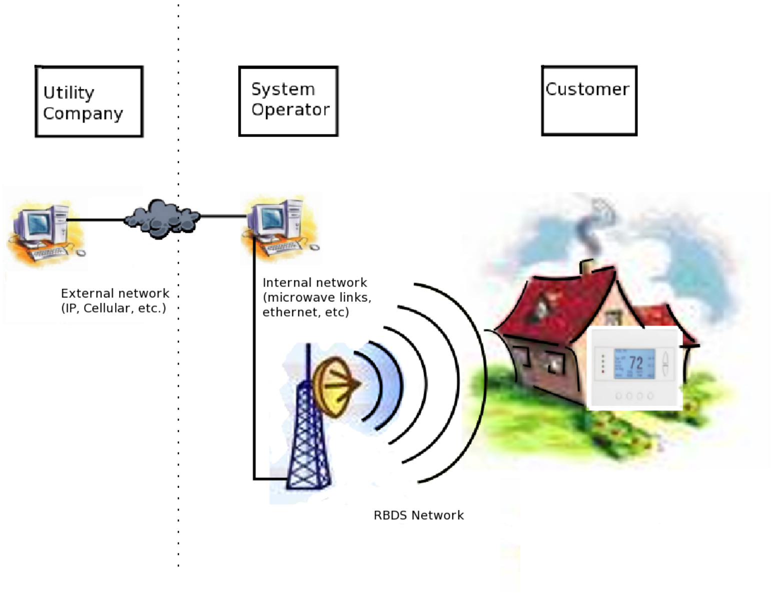

Utility Company
System Operator
Customer
External network (IP, Cellular, etc.)
Internal network (microwave links, ethernet, etc)
72
RBDS Network

Figure 4.1: Network Model for PCT System

The model of the network is shown in Figure 4.1. A utility company sends PCT messages to the network operator via an external network. The type of network used between the utility company and the network operator is agreed upon between the two parties. It could be any wide-area network. The network operator receives messages from the utility company and relays them to customers via an RBDS broadcast. The base station can be located away from the system operator. In this case, the system operator requires a link to the base station, and may use any technology (e.g. microwave links, ethernet, etc). The messages from utility companies up to and including the base station are expected to travel through tested networks, and are expected to encounter low delay and losses. A potential point of failure is the last hop from the base station to the home devices employing the RBDS network. Therefore our simulations and evaluations are focused on delivering messages over the lossy RBDS network to the home devices. Thus we simplify our model and focus on delivering the messages from the system operator to the home devices.

The characterization and evaluation of the RBDS network was done using the Network Simulator tool (NS-2.30). The communication from the systems operator to the RBDS base station uses standard NS-2 components. The communications model between the base station and the home devices uses custom models based on the RBDS network. An initial study to characterize the network necessary for comparison purposes is carried out before adding and evaluating addressing and security features. The physical layer propagation model was designed to match closely real conditions by conducting field measurements to calibrate the model. Signal strength readings of FM broadcasts were taken from various places in Ottawa and used to characterize the channel gain. Media access was also modeled to handle packet fragmentation and defragmentation to reflect the operation of RBDS.

4.1 Network Simulator 2

The Network Simulator (NS-2) tool is an object oriented, discrete event simulator written in C++. It uses OTcl as an interface for instantiating and configuring simulation objects [17]. NS was specifically designed to study networks for research and network protocol evaluation and development. The tool has been used extensively in the academic world for research purposes and hence it is the favored choice. The tool is open source and available for download at the official NS website http://www.isi.edu/nsnam/ns/. There are several websites that provide detailed starting points for using the tool. Marc Greis's website [21] gives a sound introduction that walks one through the basics of the tool. There are complementary graphical tools that allow one to analyze and visualize the simulations. Tools like Nam, Tracegraph, and Xgraph provide graphical and analytical representation for NS-2 and are also available for download.

NS-2 simulations objects are instantiated and configured through the OTcl interface by using Tcl scripts. A typical Tcl script for running an NS-2 simulation starts by instantiating a simulator object. It is through the simulator object that the simulation parameters like simulation time, dynamics and events are controlled. Files to which events of interest will be logged to are also configured in the script. A network can then be built by creating and configuring nodes and links within the simulator object. NS-2 provides a layered implementation of the communication protocol stack. This allows one to specify different implementations at each layer at each node. There are also different implementations of links and nodes from which one can choose from.

4.1.1 Changes to the Simulator

The default implementations of the physical layer, medium access and network layers in NS-2 could not be used as provided to accurately model the RBDS network.

Hence several changes to the implementations were necessary to model the RBDS network correctly. The broadcast nature of RBDS requires that the medium access (MAC) implementation be altered. It was also necessary that the routing be altered to resemble the RBDS network. The physical layer also required modifications to accurately model a wireless channel.

4.1.2 Physical Layer Model

The propagation model used to represent the wireless channel in our simulations is a combination of two models, the Shadowing model and the Ricean/Rayleigh fading model. The Shadowing model is a propagation model within NS-2 that models large scale fading of a wireless channel. The Shadowing model uses two parts, the path loss and the power deviation. The path loss part models the attenuation of the radio signal as it propagates the medium. It is characterized by the path loss exponent, which takes up different values depending on the medium conditions. The values of the path loss exponent and power deviation in practice are found empirically by field measurements. The power deviation part of the Shadowing model describes the variation of power at the receiver caused by multipath shadowing. The power deviation is described as a log-normal random variable with a mean of 0 [17]. This model allows nodes at the edge of the communication range to probabilistically receive the signal. The model effectively results in an area around the transmitter within which nodes probabilistically receive the signal. The strength of the received signal is a function of the distance from the transmitter. The model gives the power received by a receiver at a distance d from the transmitter referenced to the power at a close-in distance d_0 [17]. The equation below gives the power in dB given by the Shadowing model.

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta\log(\frac{d}{d_0}) + X_{dB} \tag{4.1}$$

The reference distance d_0 can be found by using the equation below.

$$d_0 = \frac{\lambda}{4\pi}\frac{P_tG_tG_r}{LP_r(d_0)} \tag{4.2}$$

where

P_t	is Transmitter power
G_t	Gain of Transmitter antenna
G_r	Gain of Receiver antenna
λ	wavelength of signal
L	the system loss $L \geq 1$

The Shadowing model does not take into account the time correlation of the received signal. The Ricean/Rayleigh model brings time correlation of the received signal into the physical model. The time correlation of the signal is necessary to model burst errors in the simulation. The Ricean/Rayleigh model models the small scale fading caused by movements of the transmitter, receiver and objects in the environment. The model provides a small scale fading envelope used to modulate the calculations of a large scale propagation model [31]. In this case, our large scale propagation model is the Shadowing model. The Ricean/Rayleigh fading model presents an efficient way of simulating burst errors in a packet simulator as described in [31].

The default Ricean/Rayleigh implementation uses the TwoRayGround model for the large scale propagation model. The TwoRayGround model does not provide a good representation of a wireless channel. The TwoRayGround model effectively models a circle around the transmitter within which all nodes receive the signal perfectly all the time. Hence the source code of the Ricean/Rayleigh model was modified to use the Shadowing model instead of the TwoRayGround model to represent Equation 4.3.

The two models were combined in NS-2 by combining the source code. The overall propagation model is described by the equation below:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta\log(\frac{d}{d_0}) + X_{dB} + \gamma \tag{4.3}$$

where γ is the Ricean/Rayleigh fading component.

The Ricean/Rayleigh model is characterized by the Ricean K factor, Doppler frequency and maximum velocity. The maximum velocity represents the Doppler spread caused by movements of objects in the environment. The maximum velocity variable represents the speed of receiver, transmitter or any moving objects. The Ricean distribution with K factor equal to 0 becomes the Rayleigh distribution, which models non-line-of-sight fading. For the purpose of this study, the communications is taken to be non-line-of-sight, hence the Ricean K factor is fixed at 0.

4.1.3 Calibrating the Physical Layer Model

Signal strength readings of FM broadcasts were taken from various places in Ottawa and used to characterize the channel gain. The readings were taken using a PCT radio receiver (generation 1) connected to a laptop via a serial port. The supporting software running on the laptop logs the raw RBDS groups from a frequency speci-

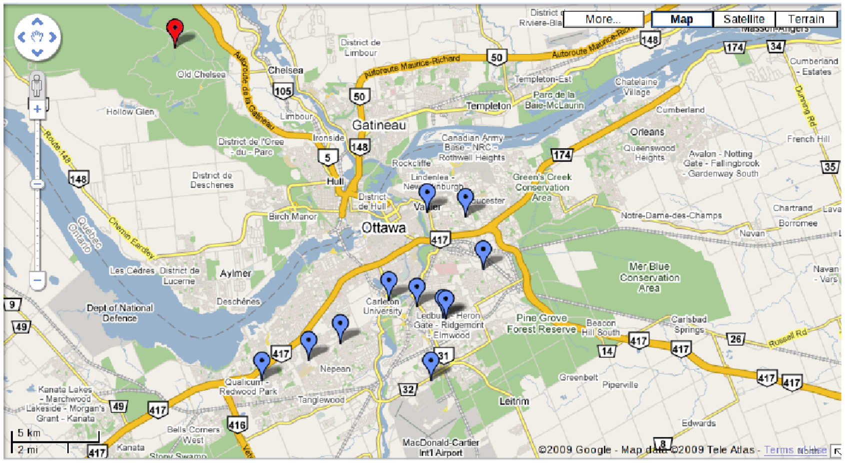


Figure 4.2: Locations of Signal Strength Measurements

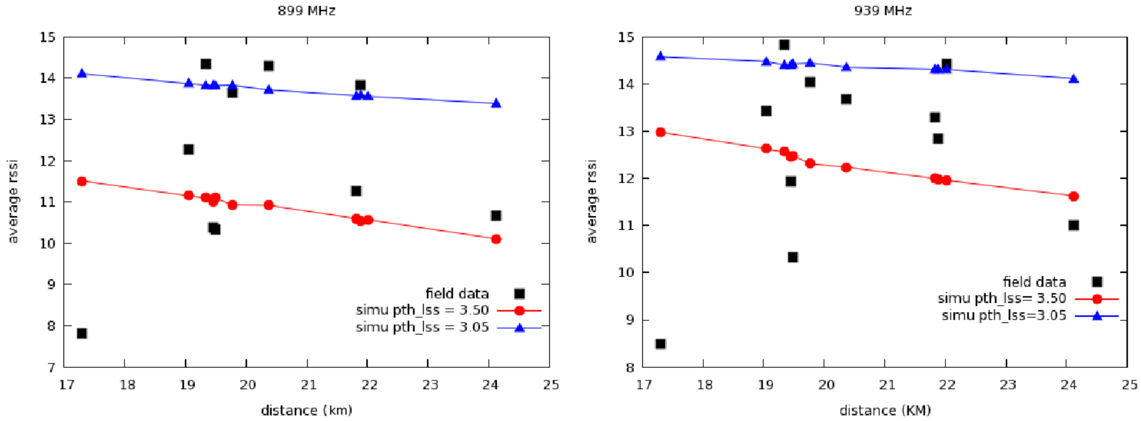


Figure 4.3: Estimation of the Communication Channel Using Field Data

fied at run-time. The logged RBDS information includes the received signal strength (RSSI), number of errors in each RBDS group, a time-stamp, and the raw bits of an RBDS group as described in Section 3.1 and shown in Figure 3.2. The locations for signal strength measurements were chosen randomly and are shown in Figure 4.2. Readings were taken from two FM broadcast channels, Hot 89.9 (operating at 89.9 MHz) and Bob FM (operating at 93.9 MHz). Both stations have their transmitting antennas in Camp Fortune, the top left (red) tag in Figure 4.2. Hot 89.9 transmits at an effective radiated power (ERP) of 27 kW while Bob FM has an ERP of 95 kW.

The signal strength measurements are presented in Figure 4.3 by squares (black). All the readings were taken indoors to reflect operation conditions for PCTs. The variations of the signal strength readings, as shown in Figure 4.3, are due to the different conditions at the measurement sites. Some of the locations were on high rise apartment buildings with potential line-of-sight conditions while others were in basement apartments. Other sites were in town houses with dry-wall and wooden walls while others had concrete walls. All these different conditions have effects on the signal propagation and hence the signal readings were diverse although they were taken from distances relatively similar from the transmission antenna. Weather conditions and other unknown environmental effects also account for the vast differences in the readings. To account for the diverse nature of the data, two approximations were made to reflect the best case and worst cases shown by the curves in Figure 4.3. Figure 4.3 also shows the estimated signal strength obtained through simulations. Two path-loss exponents were used to account for the diverse data as shown in Figure 4.3 by the two curves. Taking a conservative approach, our evaluations are conducted for the worst case scenario, hence the curve with the highest path-loss (shown by the lower curves on Figure 4.3) is used to evaluate performance. The curve with a path-loss of 3.5 forms a lower bound for the received signal strength except for two points which were taken to be outliers. Table 4.1 summarizes the physical layer parameters used for subsequent simulations.

Transmitter Power	27kW
Transmitter Height	210m
Receiver Height	1.5m
Receiver Sensitivity (Threshold)	-103dBm (-133dB); RSSI = 0
Path-loss Exponent	3.50
Shadowing Deviation	12
Ricean K Value	0.0
Ricean max Velocity	120 km/h

Table 4.1: Parameters of the Physical Layer Model After Calibration

4.1.4 Media Access

To model the media access of the RBDS network, a scheme based on TDMA was used. The default NS-2 TDMA implementation was altered to allocate all the sending time slots to the base station only. The rest of the nodes continuously listen for broadcasts from the base station in all time slots. This was done to mimic the operation of FM radios. The addressing of the packets at the MAC layer was also altered to be broadcasts. Therefore all the listening nodes receive the packets that are sent. The MAC message scheduling, fragmentation and reconstruction was modeled as described in Section 3.2.

4.1.5 Routing

Routing was deemed unnecessary in the wireless domain since all the receiving nodes are within the transmission range of the sender. Therefore the routing layer implements a dumb agent that does no routing. The DumbAgent in NS-2 was originally used for testing purposes and just passes packets between upper and lower layers. Changes made to the DumbAgent were to fix a broadcast IP address to all outgoing packets. This allows the packets to be passed up to the agents at the receiving nodes.

4.2 Simulation Results and Analysis

The reception of the signal is independent of the receiver antenna elevation over large distances from the transmitter. The distance of receivers from the base station in our models are atleast 17 km. At this distance the variation in elevation of the receiver does not have a signficant impact on the received signal. Thus the height of the receiver antenna was set at 1.5m for subsequent simulations and evaluations. The distance from the transmitter was varied and the performance of the system observed. To estimate the performance of the system, several independent replications of the simulations were run. For each setting, 10 replications were run using random predefined seeds in NS-2. The observed performance statistics are the average received signal strength and average percentage packet reception.

Simulations shows that multiple transmission of the same packets are necessary to obtain good reception of messages. To guarantee that with probability P_c, at least one packet (with probability of successful delivery p) from n re-transmissions will be received by a receiver at a fixed distance from the transmitter, the inequality given below has to be true. The calculations of the number of retransmissions assumes that the packets are lost independently to each other. Then a simple Binomial probability is used to calculate the number of re-transmissions. The number of retransmissions n is given by:

$$P_c \geq P(At\_least\_one\_success\_in\_n\_tries) = 1 - P(failure\_in\_all\_n\_tries)$$

$$= 1 - P(fail)^n$$

then

$$P_c \geq 1 - P(fail)^n$$

and

$$n \geq \frac{\ln(1-P_c)}{\ln P(fail)}$$

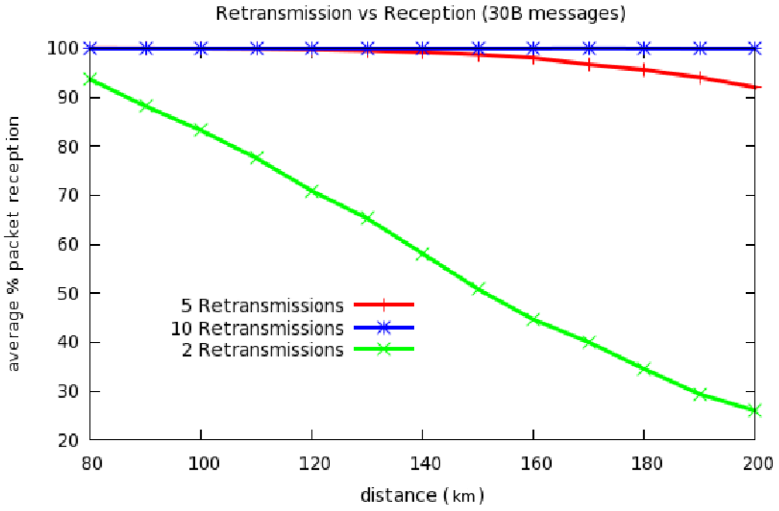


Figure 4.4: Effect of Retransmissions on Reception of Messages

Figure 4.4 shows how the number of retransmissions affects message reception. It can be seen that a message transmitted twice has less chances of reception than a message retransmitted 5 times and 10 times. A network operator needs to define a service distance and the target service level in terms of message reception probability to determine a good number for retransmissions.

The results given in Figure 4.4 are for one packet messages. For messages that consist of multiple packets, the probability of reception decreases as the number of fragments increases. For a message with m fragments and re-transmitted n times, the probability that the message is received is given by:

$$P(msg\_rcvd) = P(all\_m\_pkts\_rcvd\_at\_least\_once\_each)$$

Assuming that all packets are received independently of each other, then:

$$P(msg\_rcvd) = P(pkt_1\_rcv)P(pkt_2\_rcv)...P(pkt_m\_rcv) = [1 - P(fail)^n]^m \qquad (4.4)$$

An independent consulting study carried out by Heschong Mahone Group, Inc. for the Demand Response Research Center in Berkeley yielded the same equation [8]. Their study involved physical measurement of RBDS signal reliability in the service territory to verify that RBDS can be used for the PCT system.

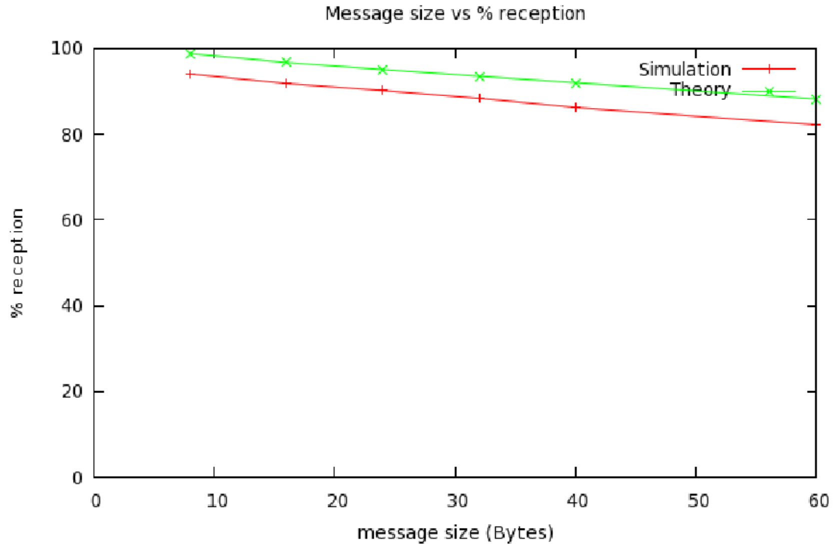


Figure 4.5: Effects of Message Size on Message Reception

Figure 4.5 shows how the message size affects the reception of messages for a fixed number of retransmissions and distance from the transmitter. The figure shows the percentage reception of messages transmitted 5 times by a receiver 120km away

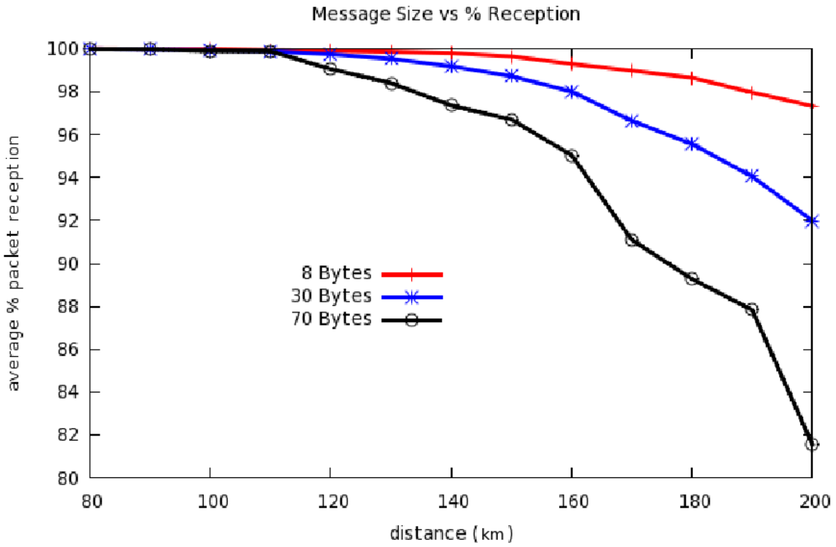


Figure 4.6: Effect of Message Size on Reception

from the transmitter. A typical transmission range target for the PCT is stated as 100 miles in [1]. It can be seen in the figure that generally a small message has a better probability of reception. The implications of this are that the messages should be small to achieve a good reception by the receivers. Simulation results are consistently worse than the theoretical calculations in Figure 4.5 because we assumed independence in reception of messages to simplify the calculations. Our simulation model uses time correlation to model burst errors, hence the messages in the simulation model are not received independent of each other [31]. Figure 4.6 shows the percentage reception of messages of differing sizes at varying distances from the transmitter. It shows how the probability of receiving messages of different sizes vary with distance from the transmitter. Small messages experience a small decrease of reception probability as opposed to larger messages. As the distance from the transmitter increases, the chances of each packet being received gets lowered. For large messages, the large value m diminishes the probability of message reception exponentially. This is shown by the fast decay in the reception of messages 70 bytes long against those 8 bytes long in Figure 4.6.

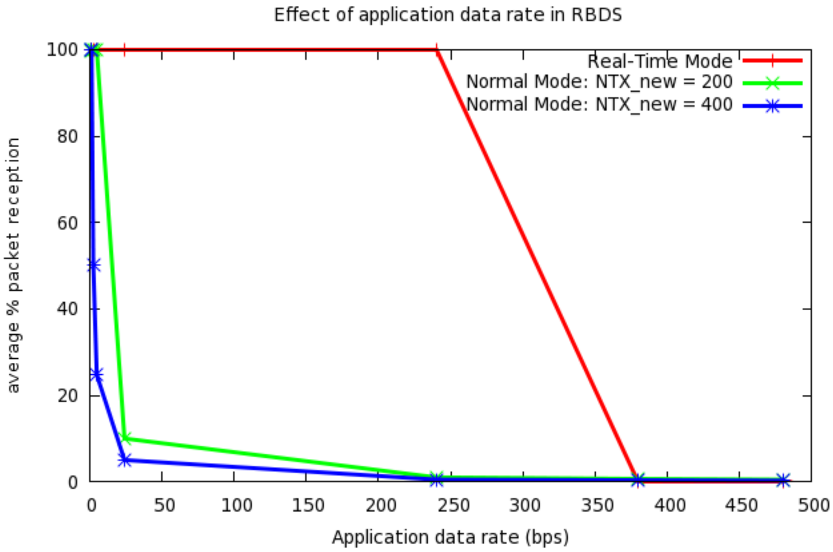


Figure 4.7: Effect of Application Data Rate on Message Reception

The dynamics of the RBDS medium access places some constraints on the application data rate. The generation of messages should not exceed the rate at which messages are sent out. Since messages are transmitted multiple times to increase the probability of reception, the time taken to send a message is long. Figure 4.7 shows how the data rate affects reception of messages. The results shown are for a receiver 120km away, with 5 retransmissions of 30 byte messages sent by the transmitter. When the data rate is smaller than 2 messages per NTX_new, the two modes of operation of RBDS achieve similar performance. As the data rate increases beyond 2 messages per NTX_new the normal mode starts to experience congestion. In the normal mode of operation, the application must maintain a data rate smaller than 2 messages per NTX_new to avoid congestion. Otherwise the messages will accumulate in the buffer until they are dropped when the buffer fills up. Figure 4.7 shows how the percentage reception of messages deteriorates as soon as the data rate exceeds 2 messages per NTX_new (for different values of NTX_new). A larger NTX_new places a tighter constraint on the application data rate as shown in Figure 4.7. The application can achieve a higher data rate for smaller values of NTX_new. If the NTX_new value is made as long as the time taken to transmit a single message, the normal mode achieves similar performance as the real-time mode. In real-time mode, the application can allow the data rate to exceed 2 messages per NTX_new as long as it remains below 1 message per $NTX\_repeat * Tx\_RBDS * MAX\_frag$ (where NTX_repeat is the number of times a message is repeated; Tx_RBDS is the time it takes to transmit a single RBDS group; MAX_frag is the maximum number of fragments for one message). If the application data rate exceeds one message per $NTX\_repeat * Tx\_RBDS * MAX\_frag$ the messages will be interrupted by new messages before all packets are transmitted.

The RBDS groups add a lot of overhead to the transmitted messages. Each RBDS group is 104 bits, of which only 32 are used for application data. Therefore the best case overhead is $\frac{104}{32} * 100\% = 325\%$. This has an impact on the effective data rate that can be achieved by the application. Assuming that the network is not used by other applications, the achievable maximum data rate is given by $\frac{1187.5bps}{3.25} = 365.38bps$. The achievable data rate reduces as the number of re-transmissions is increased to improve message reception probability. The achievable data rate shown in Figure 4.7

is 250 bps for messages with 5 retransmissions. The achievable datarate shown in 4.7 is consistent with the expected data rate of 300 bps as stated in a study carried by the University of Berkely for CEC-PIER in [1].

4.3 Conclusions

Our initial study to characterize the RBDS network confirms that RBDS is a viable option to deliver messages to home devices. The coverage of the signal makes it easy to deliver messages in both urban and rural areas. From simulations we show that devices up to 140 km can be reached with a high reception probability (above 99.5% for 30 byte messages). From the aforementioned results, it is evident that one faces tradeoffs for reception of messages. To ensure that messages are received, one needs to increase the number of re-transmission. On the other hand however, a high number of re-transmissions means that the generation rate of messages must be reduced to avoid congestion. For the the normal mode of operation of the RBDS protocol, it may be necessary to reduce the minimum time interval between successive messages (NTX_new) to be able to support high data rate applications. Otherwise the real-time mode should be used for high data rate applications.

Chapter 5

Addressing

The Radio Broadcast Data System (RBDS) has already been identified as a strong candidate technology for delivering messages in the PCT system. There is a need to provide efficient addressing schemes to meet the requirements of demand response programs over the RBDS network. The default one-way RBDS network can be overridden in the PCT system by other technologies. The addressing scheme employed for addressing devices for demand response should therefore be deployable in other networks. This chapter presents a review of addressing schemes that can be employed on the RBDS system to efficiently target devices. We then propose a scheme which could be employed to effectively target devices. The devices can be targeted by location, logical attributes, individual identifiers, or any combination of attributes. The PCT system is presented here as an example and the results can easily be used for other PCD's.

5.1 Addressing Smart Grid Applications

The problem of addressing smart pervasive grid devices is not trivial because of the number of different technologies that can be used. Currently there are few networks that completely cover the entire area of power distribution [13]. Each utility company will have a decision to make to reach all the customers in their distribution area and may have to use more than one technology. It is expected that an assortment of communications technologies (e.g. WiFi, WiMAX, Cellular,3-G networks, broadcast radio, etc) will be used to interconnect different aspects of the smart grid. To simplify the task at hand we limit the study to the default communications of the PCT system. Hence we avoid addressing based on hardware, medium access addresses, or any network specific architecture. Addressing at levels above layer 2 in the protocol decouples the addressing scheme from the underlying physical infrastructure.

Network management also needs to be taken into consideration to make the work of the network administrator as simple as possible. Tasks such as change of service point by customer relocation and change of customer profile (e.g. adding/removing customers in demand response programs, change of features of service, etc) have to be made easy for the network administrator. Different utilities other than electricity like water and natural gas distribution may have similar addressing requirements. The distribution network for such utilities may not necessarily have the same architecture as the electrical power grid so it is important not to tightly tie the addressing scheme to the electrical power distribution architecture. It is wise to allow freedom to the network administrator (responsible for naming and assigning addresses to the devices) to allow efficient use of addressing space without tight constraints.

5.1.1 Addressing Requirements

The successful implementation of demand response programs for residential use requires a flexible addressing scheme for the programmable communication devices (PCD's). A particular message sent may target an individual device, a subset of the devices or all of the devices. Emergency situations where part of the grid experience instabilities are anticipated. In such situations, the residences supplied by the troubled substation(s) can be targeted by the Systems Operator to issue an emergency event. It is important that the event message is localized to the area(s) of interest to avoid needlessly inconveniencing customers not affected by the event. Home owners may join multiple demand response programs and reward programs. Home owners in such programs will not necessarily be located in a particular geographical location, and may be dispersed over a large area. It is necessary to allow customers to be targeted by logical grouping and by locality. Instances where customers may require messages targeting an individual device also exist. Examples of such instances are the remote control of HVAC equipment by a home owner. Therefore, addressing needs to be fine enough to facilitate the targeting of individual devices and flexible to allow grouping (logical and by location) of devices.

The requirements that need to be met for addressing of programmable communication devices includes, but is not limited to, the following:

- A scalable addressing scheme that can accommodate a large number of consumers. The number of residences in a high-density urban area is in the order of a few millions, all the devices need a unique address within a network at the very least.

- A portable addressing scheme with minimal overhead is necessary for bandwidth constrained communication channels such as the RDS network. It is therefore necessary to use the addressing space efficiently.
- A flexible addressing scheme that allows easy operation and maintenance. The addressing scheme should allow location based grouping, logical grouping of devices and individual targeting of devices. Change of location, profile, and associations should be easy to perform by the operators.
- A network-independent scheme that allows communications across different platforms and networks. The messages sent to the home devices are expected to traverse different networks before reaching the target.
- A robust scheme that meets all the current requirements for demand response and allows for future changes expected in the power grid and other utilities.

The current electrical power system is expected to undergo major changes as it transforms to a smart grid. It is therefore difficult to predict and define a detailed addressing scheme that will meet all the needs of the smart grid. Distributed architectures are proposed for the smart grid which makes it very difficult to predict the final architecture of the network and requirements of devices and applications running on them. It is necessary when developing an addressing scheme to allow the changes that are expected to happen without hindrance from current practices and/or technologies. There are several addressing schemes in practice currently that can be learned from and adopted for smart grid applications.

5.2 Related Addressing Schemes

The following are existing addressing and numbering schemes that are useful for the study. The schemes present interesting insights in designing an addressing scheme for addressing home devices over a wide-area network.

5.2.1 Radio Broadcast Data System Paging

Paging services are offered in two protocols within RBDS; the Basic Paging Protocol and Enhanced Paging Protocol [2]. The Basic Paging protocol offers basic features that permit nation-wide paging services. The Enhanced Paging Protocol offers international paging services, from multiple operators and multi-area paging services. Annex M in [2] provides detailed descriptions of how both the Basic and Enhanced Paging Protocols are implemented.

Basic Paging Protocol

Paging in RBDS follows a time division multiplexing-like construct to allow receivers to save power. The transmission intervals are marked by specific RBDS group types. The transmitter transmits group type 4A which bears the clock time and date at the beginning of one minute intervals. RBDS group type 1A which carry Program Item Number (PIN) and some radio paging codes, are transmitted at least every second. Group type 7A carry the paging information.

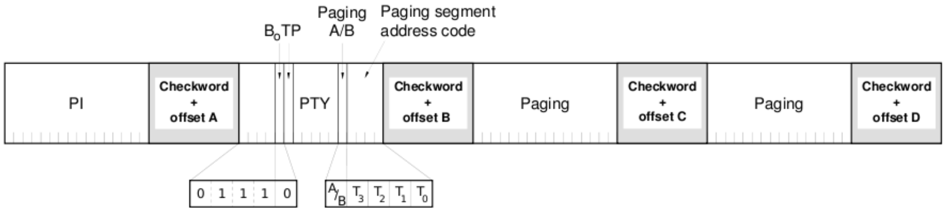


Figure 5.1: Group Type 7A Message format for Radio Paging [2]

Figure 5.1 shows the fields of RBDS group type 7A. The last five bits of block 2 (labeled A/B, T_3, T_2, T_1, T_0) are used for controlling the paging information. The bit A/B toggles its value between successive calls thus signal a repeated or a new paging call. The bits T_3-T_0 are used as a 4-bit segment address code and also serve as an indicator for the type of additional messages that follow. The type of additional messages that are possible are 10 digit and 18 digit (15 digit for international paging) numeric messages, alpha numeric messages and function messages. Only the alpha numeric messaging is presented since it is relevant to our study. The interested reader is referred to annex M of [2] for a detailed description for other message types.

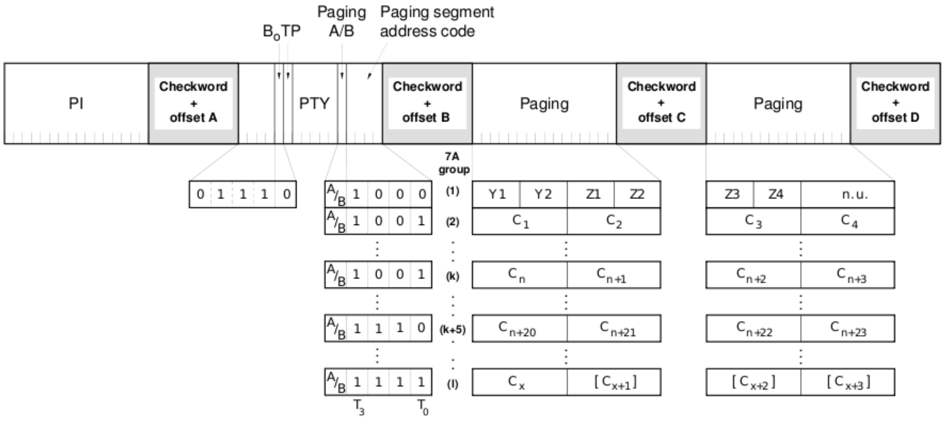


Figure 5.2: Group Type 7A for Paging Additional Alphanumeric Messages [2]

The messages characters of the message are transmitted in multiple 7A groups.

The first 7A group has a paging segment code of 1000 and carries the group and individual code (address) Y1Y2, Z1-Z4 of the target pager. Y1Y2 gives a binary coded decimal (BCD) value that denotes the group code for the target. Z1-Z4 gives a BCD value that denotes the individual code of the target within the group. The last 8 bits of the first 7A group sent are not used for basic paging, denoted n.u. (not used) in Figure 5.2. The characters of the message are then sent by successive 7A groups. The paging segments for group 7A carrying message characters range from 1001 to 1110 and repeats after every 24 characters. The paging segment 1111 signal the last 4 (or less) characters of the message. The bit A/B which signals a new call also represent the end of a message. The maximum message length is 80 characters.

5.2.2 Enhanced Paging Protocol

Enhanced paging in RBDS allows regional and international paging with multiple services and multiple operators. It also increases battery life of the pagers. With enhanced paging, it is necessary to identify the country code (using the Extended Country Code (ECC) carried in group type 1A) by the receiving pager before locking to a channel. Operator Codes (OPC) are used to facilitate multiple operators to offer paging services in the same country. Paging Area Code (PAC) defines a coverage area different from nation-wide coverage.

Paging Area Code (PAC) is defined for every country and operator. Six bits are used to define 63 paging area service areas. The value 0 for PAC translates to all the paging areas for a particular service provider (Operator). A receiver allocated the value 0 for the PAC belongs to all paging service areas and need not check the PAC information. The Operator Code (OPC) allows multiple operators to offer paging services in a country. Within a country an operator is assigned a single unique code. The OPC is made up of four bits which allows the definition of 15 distinct operators; the value 0 is not allowed.

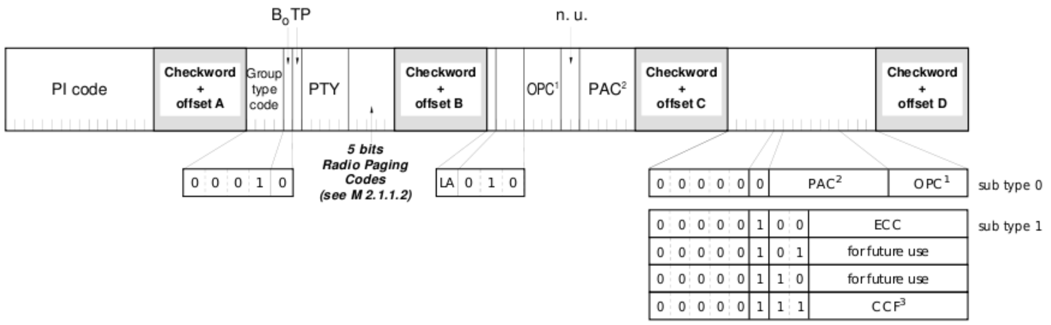


Figure 5.3: Group Type 1A Variant 2

The Paging Area Code (PAC), Operator Code (OPC), and Extended Country Code (ECC) are carried in RBDS group type 1A. Two variations of group type 1A are used, each variation has two subtypes that carry different information. The two group 1A variations used in enhanced paging are variant 0 and 2. Variant 2 is dedicated to paging and is the recommended of the two variants for enhanced paging. Figure 5.3 shows how the PAC, OPC and ECC are transmitted using group 1A variant 2. Bits 11-8 of Block 3 carry the OPC, bits 5-0 carry the PAC, while bit 7 and 6 are not used. Setting bits 15-12 of Block 4 to 0 (which usually carry the day field as part of the timing information) will cause receivers not supporting enhanced paging to ignore the rest of the block. This allows the rest of the bits of block 4 to be used to carry other information as shown in Figure 5.3. Two subtypes of variant 2 are defined as shown in Figure 5.3. Subtype 0 of variant 2 carries the OPC and PAC in Block 4. Subtype

1 carries ECC, Current Carrier Frequency (CCF), and has two options reserved for future use.

A pager is identified by a group code (Y1Y2) and an individual code (Z1Z2Z3Z4). Enhanced paging uses full hexadecimals to increase efficient use of the addressing space except for Z4, which is kept as BCD for compatibility issues. The address space with the use of hexadecimal increases from 1 million to $16^5 * 10 = 10\ 485\ 769$ addresses. The interval a pager belongs to is given by Z4 while the subgroup a pager belongs to is given by Z2Z3; hence there are 256 subgroups. Enhanced paging uses the addressing space efficiently and does not place a restriction on message size. However, it is recommended that messages not exceed 80 characters. Enhanced paging allows variable function messages to be sent to pagers, which, as an example, could be used to program the pagers.

5.2.3 IEEE Utility Communications Architecture V.2

The IEEE Utility Communications Architecture (UCA) uses network service access points (NSAP) addresses to unambiguously identify end devices within a network. In the UCA, an NSAP address is required to be unique within a network and globally if the network is connected to other networks [3]. The UCA NSAP address is hierarchical as described in IEEE-SA TR 1550-1999 [3] and has three levels of hierarchy as shown in Figure 5.4. The authority and format identifier (AFI) identifies the first level of addressing authority and the syntax (binary or decimal) of the domain specific part (DSP) of the address. The first level authorities include ISO standards and

ITU recommendations such as ISO 3166-1:1997 which is identified by a value of 39 in the AFI field. The initial domain identifiers (IDI) identifies a second level authority recognized by the authority specified by the AFI. The ISO 3166-1:1997 assigns the each country an IDI value and within each country the local address agency acts as the second level of authority. In the USA (IDI of 840) the second level (or local) authority is the American National Standards Institute (ANSI). The contents and structure of the DSP field are not specified by ISO 8348:1996 or by ANSI.

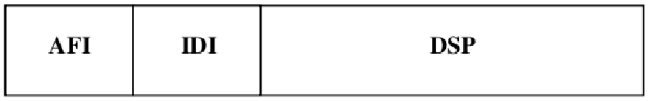
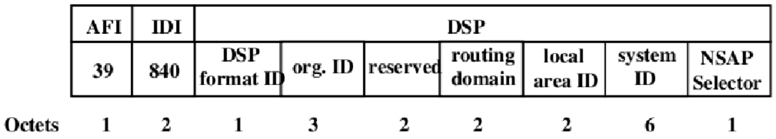


Figure 5.4: General Structure of a UCA NSAP Address [3]

UCA NSAP GOSIP Format

The GOSIP address format is 20 bytes long and addresses the DSP field in a hierarchical manner to make for efficient routing [3]. The structure of the UCA NSAP GOSIP address is shown in Figure 5.5. The reserved field is defined to allow for future extensions to the NSAP address. The routing domain, local area ID and system ID fields are used to reflect the structure that the organization has. The routing domain unambiguously identifies a routing domain within the network. The local area ID identifies a subnetwork or area within the routing domain while the system ID identifies a particular system with the local subnetwork or area. The NSAP Selector identifies a user of the network service (e.g. transport service) within the end system.



	AFI	IDI	DSP						
	39	840	DSP format ID	org. ID	reserved	routing domain	local area ID	system ID	NSAP Selector
Octets	1	2	1	3	2	2	2	6	1

Figure 5.5: Structure of the UCA NSAP GOSIP Address [3]

IP based CNLP addresses

The use of IP-based connectionless network layer protocol (CNLP) addresses allow utilities to use IP addresses and networks using the IP private address space. The use of an IP address produces a compact address with less overhead than the UCA NSAP GOSIP address. A global IP address from CNLP uses 14 Bytes instead of 20 for GOSIP. The fields of the address are as given below:

AFI: 39 (1 byte)

IDI: 840 (2 bytes)

DSP: format (1 byte)

Org ID: Organization ID (3 bytes)

Local Address: 10.xx.xx.xx (4 bytes)

Sel: NSAP Selector (1 byte)

In some cases the organization does not require that the devices have global visibility. In this case the address field can be made even more compact. The network in this case carries internal traffic only and the address can be reduced to three fields as shown below:

AFI: 39 (1 byte)

Local Address: 10.xx.xx.xx (4 bytes)

Sel: NSAP Selector (1 byte)

5.2.4 ITU-T Recommendation E.164

The ITU-T Recommendation E.164 specifies the addressing structure and functionality for the four major categories used in international public telecommunications: geographic areas, global services, networks and groups of countries. The E.164 recommendation provides for global addressing for both fixed and mobile terminals. The recommendation explains the components of the numbering structure and call routing. An international E.164 number is specified to be at most 15 digits long in the recommendation. An international E.164 number is made of two variable length fields: the Country Code (CC), and the National Significant Number (NSN).

An E.164-number for geographical areas is shown in Figure 5.6. In the case of a geographic area E.164-number, the NSN is made up of two parts: National Destination Code (NDC) and a Subscriber Code (SN). The NDC is specified as a variable length field and depends on the requirements of the network. It is composed of any combination of Destination Network (DN) code and a Trunk Code (TC). The DN and TC combination and sequence is determined by the requirements of the country and may be different in different countries. National numbering plan administrators (e.g. ANSI) determine the DN and TC combination and sequencing. The SN also varies in length and depends on the requirements of the country.

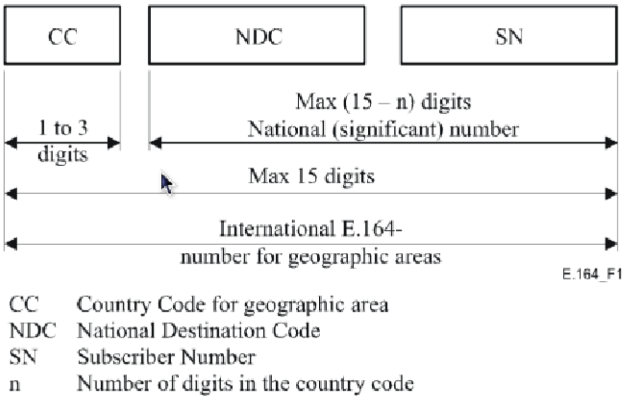

CC
NDC
SN
1 to 3 digits
Max (15 – n) digits
National (significant) number
Max 15 digits
International E.164-number for geographic areas
E.164_F1
CC Country Code for geographic area
NDC National Destination Code
SN Subscriber Number
n Number of digits in the country code

Figure 5.6: International E.164 Number Structure for Geographical Areas [3]

The ITU-T E.164-number for geographical areas is used in land mobile communications like GSM networks [4]. The numbering structure in Terrestrial Trunked RAdio (TETRA) networks used in mobile communications is similar to the international E.164-number for geographical areas [4] [5]. An Individual TETRA Subscriber Identity (ITSI) is shown in Figure 5.7. The fields shown in the ITSI address structure are equivalent in functionality to the fields in an E.164-number. The TETRA Mobile Country Code ((T)MCC) is equivalent to the Country Code (CC) of the ITU recommendation. Similarly, the TETRA Mobile Network Code ((T)MNC) and Subscriber Identifier (SSI) are equivalent to the National Destination Code (NDC) and Subscriber Number (SN) fields in the National (Significant) Number (N(S)N) of the ITU recommendation.

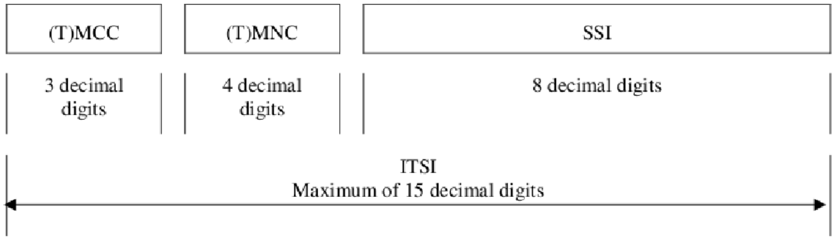


Figure 5.7: Structure of an ITSI Address [3]

An ITU-T E.164-number for global services is made of two parts: a three digit Country Code (CC) and a variable length Global Subscriber Number (GSN). The structure of the ITU-T E.164-number is shown by Figure 5.8. The CC for global services is used to identify the global service and is 3 digits long (e.g. + 800 - XXXXXXX). The GSN is variable length and depends on the requirements of the service, e.g. 911 service does not require a GSN (i.e. GSN is length 0) while toll free service (+ 800) requires a GSN (usually 7 digits).

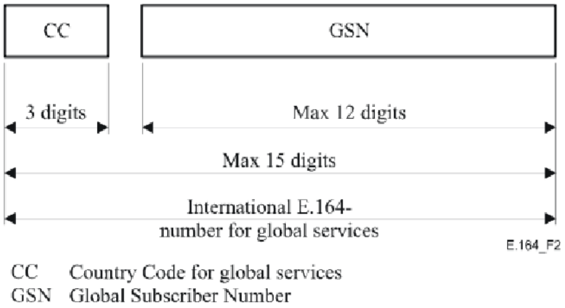
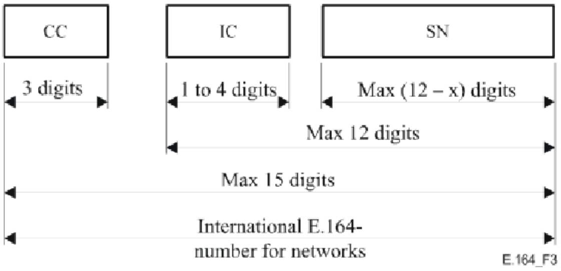

CC
GSN
3 digits
Max 12 digits
Max 15 digits
International E.164-number for global services
E.164_F2
CC Country Code for global services
GSN Global Subscriber Number

Figure 5.8: International E.164 Number Structure for Global Services [3]

The international E.164-number for Networks is made up of a three digit Country Code (CC), and variable length Identification Code (IC) and Subscriber Number (SN). The structure of the international E.164-number for networks is shown in Figure 5.9. The Country Code (CC) for networks is used together with the Identification Code (IC) to identify a network. The Subscriber Number length, structure and functionality is dependent on requirements and is determined by the network operator.



CC Country Code for Networks
IC Identification Code
SN Subscriber Number
x Number of digits in Identification Code

Figure 5.9: International E.164 Number Structure for Networks

5.2.5 Summary

The addressing schemes reviewed in the literature have common drawbacks that make them non-ideal for addressing devices for demand response use. Their use for addressing devices grouped in various flexible ways for demand response applications is limited. The addressing schemes were designed to address devices within a single static group. The groups used in demand response are expected to be dynamic, e.g. customers may change demand response programs, location or profiles. Therefore, a more flexible grouping scheme is required to address devices based on location and logically. The addresses in the presented schemes are large and will result in a large communication overhead when sending demand response messages, which are expected to be small (i.e. in the order of a few tens of bytes). The UCA NSAP GOSIP address is 20 bytes long and will create a 66.67% overhead for demand response messages 30 bytes long. The E.164-number is 7.5 bytes long (accounts for 25% overhead for a 30 byte message) and the enhanced paging address is 3 bytes long (10% overhead for a 30 byte message). According to the initial study, larger messages will experience a higher loss rate. Hence it is preferable to use an addressing scheme that is bandwidth efficient. We present and propose an addressing scheme for use in residential demand response programs in the following section.

5.3 Proposed Addressing Scheme for Smart Grid Applications

There is no single right answer to solving the problem of addressing the devices in a smart grid. After examining the existing addressing schemes available we propose an addressing scheme based on the preceding addressing schemes. All the presented addressing schemes group devices and then identify an individual device within a group. The addressing schemes however are primarily designed to identify a particular device not a group of devices. We therefore propose a variable length address structure to allow addresses to be as compact as possible and only as large as they are required to be. The primary requirements that we identify above are met by using flexible addresses.

For demand response applications we assume there will not be international communications of pricing events. Hence it suffices to appoint a national numbering or addressing authority to assign and manage addresses, numbers or codes for utility companies. A field in the addressing scheme that identifies a territory is necessary for inter-territorial communications. Inter-territorial communications is also not expected to be prevalent for demand response applications, but it may be necessary for other smart grid applications and emergency services. Within a territory there may be several utility companies operating and will require to be distinguished from one another. Figure 5.10 shows the structure proposed for addressing home devices for smart grid demand response applications.

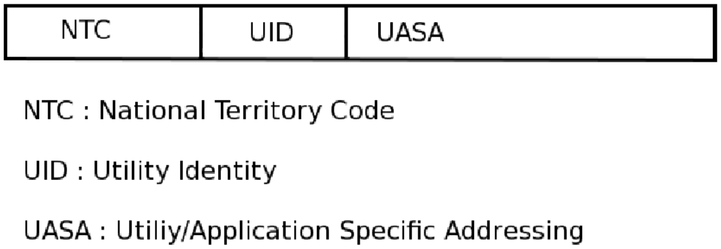

NTC
UID
UASA
NTC : National Territory Code
UID : Utility Identity
UASA : Utiliy/Application Specific Addressing

Figure 5.10: General Structure for Smart Grid Addresses

The National Territory Code (NTC) identifies a specific region in the country. The Utility Identifier (UID) identifies a utility company within a territory that offers the service. Both NTC and UID should be assigned by a national numbering or addressing authority. The Utility or Application Specific Addressing (UASA) addresses the target device or devices within the distribution territory. It is variable length, which is determined by the network operator to meet the requirements of the applications and services. The network operator has the flexibility to structure the UASC as required by the major applications or the needs and structure of the distribution architecture of the utility.

For the use of electrical demand response in residential devices employing the RBDS network it is important to note that only local traffic will be carried in the network. The National Territory Code (NTC) can be omitted while communications are limited to local traffic only. The UID and UASC are sufficient to unambiguously target devices locally. We want to allow the network operator sufficient freedom to group customers effectively and efficiently. For residential demand response, most of the communications will be multicasts (e.g. pricing events, emergency events). Therefore, the addresses by default target a group. This allows the addresses to be compact. Where necessary, the address can be extended to target a particular device. In this case, additional information identifying a device within a group will be required. Thus our proposed scheme is variable length to allow efficient bandwidth use.

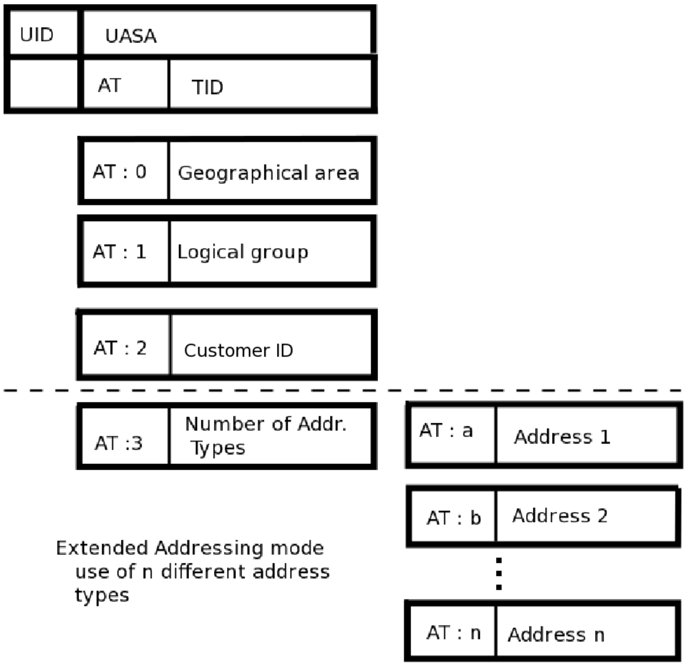


Figure 5.11: Proposed Structure for Addressing PCT's in RBDS Network

We propose that the UASC has two variable length parts: Address Type (AT) and Target ID (TID). The Address Type specifies the type of address (e.g. logical, geographic, and extended) carried in the Target ID field. A logical Address Type means that the addressee (identified by the TID) is a logical group e.g. a certain rewards program or customers with certain profile attributes. A geographic Address

Type specifies that the target is a geographical area e.g. a section of a city supplied by a certain substation or feeder. There may be instances when specific devices are targeted, e.g. thermostats (for PCT system), water heaters, etc. An address type for devices may be required. There are a number of possible address types that could be targeted. New applications may also require address types that cannot be predicted currently. To allow for such developments we recommend sufficient flexibility be allowed by reserving some options for future use in the Address Type. It may also be necessary to have a hybrid of two or more address types, e.g. customers of logical group A in a geographical area X. In this case the Address Type specifies the Target ID as a combination of two addresses. On a general point, there may be more address types that arise because of new applications and services. These may be combined with each other to target devices. To allow the use of multiple address types, the extended Address Type is used to specify how many address types are used. The addresses are then sent sequentially to specify the target as the intersection of all the address types that make up the address. The order and structure of multiple address types is left to the network operator to efficiently determine. Targeting an individual device is also allowed, the target in this case a specific device. Again here, the network operator and utility company have the freedom to use any means they see effective, e.g. account number of customer, serial number of meter, customer phone number etc, can be used as a master address for a user. Then a device within the user's home can be identified. Figure 5.11 shows the structure of addressing PCT's over the RBDS network. It is important to notice that the logical group and geographical area address types can be made as small as possible to make the addresses compact. These modes are anticipated to account for the majority of the traffic sent to the PCT's hence bandwidth can be used efficiently. The lengths of both the AT and TID fields will be determined by the requirements and anticipated future needs of the network

An example of electrical demand response program in a city with up to 10 million homes employing RBDS to deliver messages is presented below to demonstrate usage of the proposed addressing scheme. We assume there could be up to 64 utility companies in a territory, therefore 6 addressing bits can be used to uniquely identify each utility company (UID) in the structure shown in Figure 5.11. We reserve 5 bits for the Address Type field to distinguish up to 32 different address types to allow future application to be integrated into the system. Each customer can have a number of devices that can be used for demand response. Each of the devices can be targeted by the utility to send device specific messages. We therefore have a master address identifying a customer or home. Each customer or home is assumed to have no more than 16 addressable demand response enabled devices. Thus 4 addressing bits are enough to identify the devices within a home. As an example, device number 0 could be assigned to all devices, device number 1 assigned to PCTs, device 2 to water heaters, etc. To uniquely address each home, 24 addressing bits can be used to identify 16 777 216 users uniquely. The address over the RBDS network therefore could be 40 bits long (we added an extra unused bit to make the size a nice round number) and requires 2 RBDS groups to send to the receivers. This address scheme will only be used occasionally when messages target a particular customer or home as shown in Figure 5.12. As previously explained, the UID selects a utility company, the Address Type, Customer ID and Device number form the UASC. The Address Type in this case shows that the address targets a specific customer, the Customer ID identifies the customer while the Device number selects a particular device at the customer's home.

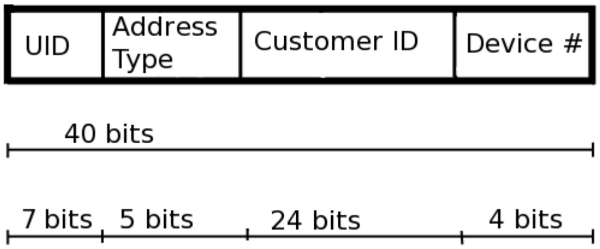


Figure 5.12: Address Targeting a Customer Over RBDS

It is more likely that most of the messages targeting a particular device type, say PCTs, will be targeting either a logical or geographical group. As such, the grouping addresses can be augmented with bits reserved for specific devices. Thus, if there are a total of 100 different demand response programs (logical groups), then 7 bits to identify a total of 128 different demand response programs, and 4 bits to identify a device within that group can be used. In this case the address to target devices with a logical group can be achieved with 11 addressing bits. The total address including the UID (6 bits) and AT(5 bits) is 22 bits which van be carried in a single RBDS group. Similarly for the geographical areas, the target could be a specific device type within a given area (e.g all water heaters in area A). Figure 5.13 shows the structure of an address used to target a group over the RBDS network.

When necessary, the extended mode of addressing can be used to form complex grouping of devices using a series of group addresses. For example, a message could

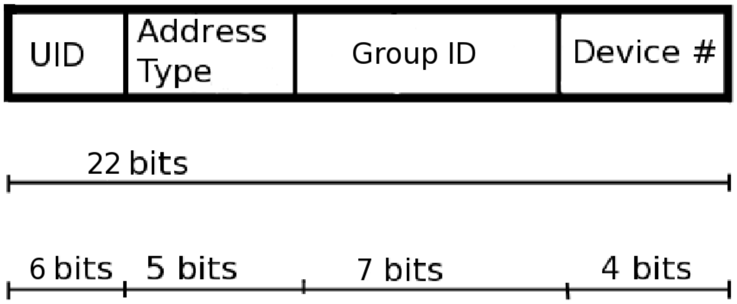
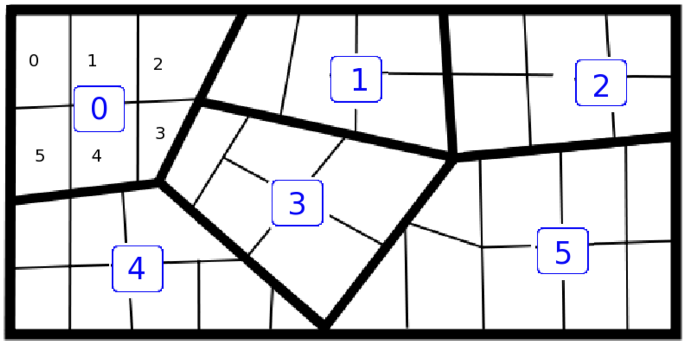


Figure 5.13: Addressing Targeting a Group Over RBDS

target every device X, within a geographical area Y, whose owners are in demand response program Z. In this case two addresses will be sent sequentially, one identifying the device type and location, the other identifying the demand response program.

The impact of the address overhead is minimal in this case and thus it is expected to have minimal effect on the reception of messages. The reception probability of messages in the initial study was shown to follow Equation 4.4. Therefore for a message 30 bytes long ($m = 30/4 = 8$ RBDS groups), the address overhead will be on average 1 or 2 RBDS groups with each of the n re-transmissions. The probability of reception of the message will be 0.999996875 with the addressing overhead against 0.9999975 with no addressing overhead, for a 95% RBDS group reception rate (i.e. $P(fail) = 0.05$). The difference in reception probability is $6.24994 * 10^{-7}$. The difference is so small that simulations do not yield significant differences.

We recommend a hierarchical numbering or naming of geographical areas similar to that used in postal codes to allow efficient grouping of devices. A hierarchical numbering scheme of geographical areas allows adjacent geographical areas to be grouped into one single group. A flat numbering scheme would require sending multiple messages to each of the targeted groups even if they are adjacent. An example of how the numbering plan of geographical areas can be achieved is illustrated by Figure 5.14. The figure shows two levels of hierarchy. Level 1 can be used to address primary substation in an area and level 2 used to identify secondary substations.



msk1	level 1	msk2	level 2

Address Structure

Figure 5.14: Numbering of Geographical Areas by a Utility

The structure of the geographical area address is shown at the bottom of Figure 5.14. The address has 2 mask bits (msk1 and msk2) which signal if an address field is important or not. If all bit masks are set to zeros, then all devices in the entire area receive the message. If the bit mask for level 1 (msk1) is set and the level 2 bit mask is not set, then all devices within area addressed by level 1 will receive the message. This case represents all the residences that are served by a particular primary substation. If all the bitmasks are set, then the only devices addressed by both level 1 and 2 will receive the message. This case targets all residences that are served by a particular secondary substation. In the area shown in Figure 5.14, an area represented by a level 1 address of 0 will be all the devices in the top left hand section of the area (which is supplied by a primary substation number 0). Within the area, a number of secondary substation numbered 0 through 5 supply residences. To target residences serviced by secondary substation 3 within area 0 for example, one specifies level 1 field as 0 and level 2 field as 3 and sets both masks to 1. Grouping areas this way targets devices more efficiently than in the case of a flat structure. For example, if a flat structure is used in an area such as the one depicted in Figure 5.14, it would require 6 individual messages to target the area addressed as 0. With the hierarchical structure, one message with a value 0 for level 1, msk1 bit set to 1, and msk2 bit set to 0 is sufficient. The logical groups addresses can similarly be assigned in such a manner if the logical groups can be arranged as subsets of other groups.

5.4 Conclusions

Addressing of pervasive devices has been examined to identify concepts that can be adopted to deliver demand response message to home devices for smart grid applications. Our work proposes a compact and efficient addressing mechanism that allows flexible addressing of devices based on locality and logical association of devices. The proposed geographical area address allocation makes for efficient use of limited bandwidth by allowing small and large groups of devices to be targeted easily. The addressing scheme allows each network operator a great degree of freedom to name and address devices efficiently and it is not strictly limited to a particular network technology.

Chapter 6

Security Performance

6.1 Introduction

The wireless nature of the communication infrastructure puts the smart grid applications running over it at a security risk. The RBDS network does not employ any security mechanisms on which the PCT or a PCD system can rely. Therefore there is need to provide for secure means of communicating PCD system messages over the RBDS network. In the PCT system, privacy is not as much a priority as authentication. The event messages are to be broadcast to alert everybody about events, therefore there is no need to make such messages secret. Authentication however is necessary to ensure that only authenticated messages are responded to. As pointed out in [7], an attacker could cancel events prior to their intended period elapses. The attacker in this and many other ways can cause distress and possibly cause grid instabilities, effectively defeating the whole purpose of demand response. Therefore, the PCTs have to authenticate the origin of the message and only react to messages originating from an authenticated sender(s).

We propose solutions to address the security issue over the RBDS network. The solutions presented here can be employed to authenticate messages sent by any application using the RBDS network as the physical infrastructure. In [7], the possible security threats to the PCT system are studied and a risk management approach is used to propose mitigation steps for the security concerns. Our study provides an analysis of the security threats for a communication protocol for use with PCTs over the RBDS network. A literature survey of security issues in similar networks is carried out to identify solutions that could be used or extended to the PCT system. Of particular interest are sensor networks and RFID networks because they face similar challenges of limited resources. We also identify possible solutions that could be pursued to provide authentication over the RBDS network and their impact on the network. Three authentication schemes identified to be suitable for the RBDS network, (BiBa, HORSE, ECDSA) are investigated using simulations to determine the impact on network resources.

There is need to provide for security in the design of the communication protocol as recommended by [7]. The security of such a system should be resilient to attacks and be able to recover easily from a breach. Bono *et al* show in [34] that obscurity is not a good measure for ensuring security. They advocate the use of standard cryptographic algorithms employing keys of sufficient lengths. They demonstrate this by bypassing the immobilizer of a vehicle which employs a cryptographically-enabled RFID tag. They achieved this by reverse engineering, key-cracking and simulation. The immobilizer in their study employed a Texas Instrument Digital

Signature Transponder (DST). From the knowledge of a rough schematic posted on the Internet, they were able to determine the functional details of the cipher of the DST. The challenge/response authentication messages between the reader and the tag were obtained and used to crack the key. The 40-bit shared secret key was extracted with the use of an array of FPGAs in less than an hour. Then, using the extracted key, they were able to simulate the RF output to spoof the reader. In their study, they were able to establish conditions for hot-wiring a car with fairly modest resources.

Strong cryptographic algorithms add to the complexity and ultimately the cost of manufacturing the devices. The price of the PCTs has to be minimized as they are expected to retail at less than $50 [7]. Sensor networks and RFID networks face similar problems with the need to provide security and still keeping the cost of the devices relatively low. It could be expected that the PCTs may have slightly more computing resources, storage, and power supply than RFID tags and sensor nodes. However, the PCTs are still expected to have modest computing and storage resources compared to today's computers. This limitation means that the security and authentication algorithms employed on these devices be efficient and low cost.

An initial study of the security characteristics of the PCT system in [7] advocates a tiered security solution. The solution defines the System Owner as responsible for overseeing and controlling the PCT system. All the messages that the System Owner sends to the PCTs go through the System Operator. The System Operator is responsible for delivering the messages to the PCTs within its geographical or logical coverage area. The goal of our study is to provide secure communication between the System Operator and the PCTs using the RBDS network.

Authentication poses more of a challenge in a one-way communication channel because conventional authentication methods of challenge/response cannot be used. In a challenge/response authentication, a sender proves its identity to the receiver by responding to a challenge from the receiver and vice versa. This cannot be done over the RBDS network, since the PCTs do not have a communication channel to the System Operator with which they can challenge the identity. Even if such a channel existed, the volume of challenges coming from the PCTs would be too high to make this approach attractive.

6.2 Threat Model

The characteristics of an adversary and the impact of the threat posed need to be established before discussing mitigation strategies. The PCT system is subject to a number of attacks as stated in [7]. The adversary that we discuss in our study is limited to one who attacks the PCT system via the wireless communication channel. The motives of such attackers could be anything from leisurely mischief to a terror attack targeting denial of utility services to customers. According to [7], the attacks that an adversary could launch on the PCT system include, but are not limited to, the following :

- An attacker could cause unanticipated loads on the grid causing instabilities by sending false messages to customers. This could be done by canceling valid emergency event messages aimed at alleviating existing grid instabilities thus preventing the expected reduction in load.

- An attacker could send false time synchronization messages creating erroneous behavior of the PCTs.

- Customers could be deceived by false messages displayed by the PCTs if an attacker can successfully send such messages to the PCTs.

- A successful breach of the communication can allow an attacker to shut down PCTs or even install new software into the devices. An attacker who is able to shut down PCTs remotely could cause irritation, discomfort and health problem to some users. The installation of new software (potentially malicious) by an attacker may lead to erroneous operation of the PCTs.

- An attacker could jam the signal to a subset of receivers from a ground station or aircraft e.g. balloon.

A systematic risk analysis of the threats posed to the PCT system and counter measures is fully described in [7]. For the purpose of our study we address the threat and mitigation procedure for a PCT system employing the RBDS network to communicate messages.

The nature of the RBDS network limits the way an attacker can launch attacks. The lack of a reverse communication channel from the PCTs to the System Operator means that the attacker should have physical access to a PCT to access information on it. We assume that an attacker has unlimited access to PCTs, either from his/her own home or he/she could break into someone's home and access a networked PCT. Moreover, an attacker could easily purchase the device at a retail store. This means that the data stored on these devices can be retrieved by a determined attacker using any method at his/her disposal. This setting does not bode well for security by obscurity of cryptographic keys. We assume it would be fairly easy for an attacker to retrieve a decryption key(s) from the PCT, thus the use of symmetric key cryptography should be avoided. Asymmetric cryptographic methods are more favorable for this setting. If public-key cryptography is used, an attacker would only retrieve public keys of the System Operator by attacking the PCTs. An attacker would be forced to attack the Systems Operator to obtain the private keys that would allow him/her to encrypt messages.

Although the attacks on the PCTs are easy, as mentioned above, attacks on the sender (System Operator) are not trivial. An attacker wishing to get information from the sender is limited to eavesdropping or gaining direct (or indirect) access to the system information database. The former method of attack means that the attacker is limited to what is communicated and what is stored on the PCTs. Methods of communication that reveal no information to an eavesdropper and store no critical information at the receiver should be employed to lower the risks of this type of attack. Gaining physical access to the system database is not easy but none-the-less possible. An adversary could break into the premises and obtain critical cryptographic information that would allow him/her to launch an attack. The critical information used for communication could also be leaked through employees to an adversary by negligence, blackmail, extortion or ignorance, to mention a few. A decentralized method of storing cryptographical information should be used to avoid a single point of compromise. To protect the cryptographical information, [7] suggests that complementary pieces of cryptographic information should not be stored in one place or exposed to one person.

RBDS works in a synchronized manner with the receiver being periodically updated by timing information from the base station. This property is exploited to avert attacks on the system where by an attacker sends messages with erroneous timing and/or replaying some messages. The application data include timestamps and message sequence numbers that make it hard for an attacker to launch replay attacks or to game the system [7]. Each message is timestamped, which associates it with a particular point in time. The devices upon reception of a message verifies the timestamp based on synchronization of sender and receiver. The message identifiers employs values similar to cryptographic nonces. This means that an attacker cannot predict or guess subsequent message identifiers after receiving a valid message. A receiver has a way of quickly verifying if a message is being replayed by checking with a small subset of message identifiers that it stores upon receiving messages. This structure makes replay attacks difficult to execute even without any cryptographic security measures. This is because if a receiver receives a message with an identifier that has been received, it discards such a message. If this construct is coupled with cryptographic measures like digital signatures, two message sent at different times will have different signatures (due to the timestamp and random message identifier which form part of the message) even if they issue the same command. An adversary who does not know the private key (assuming an asymmetric cryptographic security protocol) will be unable to generate a valid signature for subsequent messages even if he was able to guess the next random message identifiers. The use of cryptography for source authentication implicitly makes it even harder for an attacker to carry out replay attacks and such attacks will not be considered in the following discussions.

Operation on the FM radio spectrum requires licensing from the radio spectrum management organization. An attacker operating unlawfully on a frequency without a license would be stopped if detected. As part of their non-cryptographic solution to provide security, [7] proposes the use of monitors placed to detect infringements. The monitoring devices should be conveniently placed to receive the messages and compare them with those sent by the System Operator. These devices, carefully placed in the coverage area, will reflect what the PCTs receive from the network. With such measures in place, it would be easy to detect if the messages are tampered with or if there are new unaccounted messages showing up at the PCTs. The authorities then would be alerted of the infringement. The use of a detection system cannot be relied upon to provide security. The attacks on the PCTs can go unnoticed if they are targeted to a small subset of customers who are in the blindspot of the monitors. Moreover, the attacker could be mobile and operate for a short period and leave no trace. In such cases it would be very difficult for authorities to stop future attacks by the same attacker even if such attacks are detected.

6.3 Literature Survey on Possible Mitigation Steps

Several symmetric and asymmetric cryptographic methods exist in the literature. A survey that covers the technical problems faced by RFID security and privacy is presented in [27]. Several cryptographic methods are proposed in the literature for RFID tags in [12] and [18]. The methods of cryptography favored by the security experts consulted in [7] is the use of Elliptic Curve Cryptography (ECC). The following methods were identified as promising to the application for electrical demand response in residential devices. Further studies pointed out shortcomings that made them non-ideal for RBDS setting.

6.3.1 Secure DNP3

Mander *et al* in [36] discuss a distributed security architecture using the Distributed Network Protocol (DNP3) to offer security for residential load-management devices. Their solution protects Intelligent Electronic Devices (IEDs) networked to a SCADA network from cyber attacks. The DNP3 protocol is used widely in the world for electricity and water utilities for communications with field equipment [20].

Figures 6.1 and 6.2 show how NDP3 is used for authenticating application messages between two entities. DNP3 employs the challenge/response authentication method to authenticate critical commands as shown in Figure 6.1. The receiver queries the identity of the sender upon receiving a critical command. The sender proves its identity by demonstration of knowledge of shared cryptographic keys. The aggressive mode as shown in Figure 6.2 is used to conserve bandwidth by eliminating the

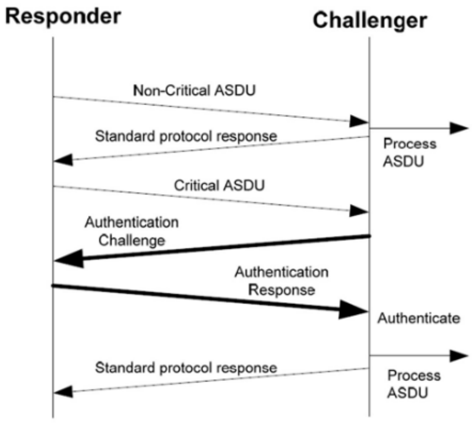


Figure 6.1: DNP3 Challenge Response Mode

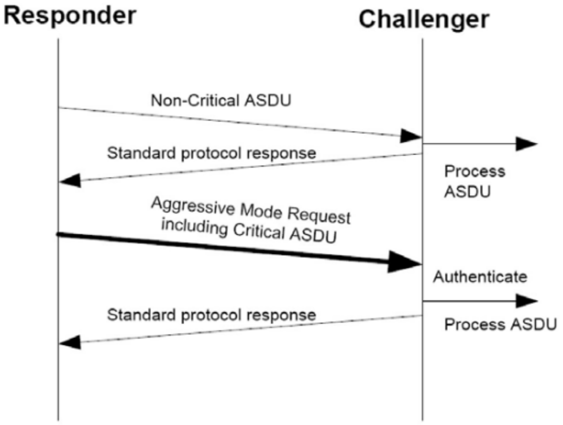

Responder
Challenger
Non-Critical ASDU
Standard protocol response
Process ASDU
Aggressive Mode Request including Critical ASDU
Authenticate
Standard protocol response
Process ASDU

Figure 6.2: DNP3 Aggressive Mode

challenge and response messages. In the aggressive mode, the authentication data is included in the message. The aggressive mode is considered slightly less secure than the normal challenge/response mode [20]. For the aggressive mode to be used there has to be at least one request/response authentication preceding it to establish trust between communicating entities. The DNP3 method of authentication cannot be completed over the RBDS network because of the lack of a reverse channel for home devices to challenge the identity of the sender.

6.3.2 Authentication Using RF Fingerprints

The physical layer RF fingerprints can be used together with higher layer protocol methods to provide authentication [15]. RF fingerprints identify an RF transmitter from the properties of the received radio signals. They allow different transmitters to be distinguishable from one another. Home devices could be enabled to do RF fingerprinting and use location in the credentials of the sender (System Operator). In this way an attacker masquerading as the System Operator would be forced to operate very close to the legitimate System Operator. If an attacker is forced to operate near the legitimate System Operator, then his/her effective radiated power would have to be comparable to the System Operator for his/her signal to be detectable. The total effective radiated power for systems operating in the RBDS network are in the order of tens of kilowatts. The cost of equipment and operation should serve as a deterrent for most attackers. Even if the attacker was able to obtain the equipment and broadcast messages, the spectrum management regulation body monitoring the use of the radio spectrum could be relied upon stop the unlawful operation.

The home devices (PCTs) should be able to learn the new fingerprints of the System Operator if the transmission RF equipment changes for any reason. The PCTs should be able to distinguish an attacker from a legitimate System Operator with changed fingerprints. Additional hardware and digital signal processing (DSP) units would have to be incorporated into the PCTs to enable RF fingerprinting. The extra hardware could potentially drive the cost of the PCTs high. The cost of such additional hardware is unknown to the author and requires investigation to determine if the solution is cost effective.

6.3.3 Zero-Knowledge Device Authentication

A method that allows pre-authenticated response between RFID tags and readers was presented in [33]. The solution curbs divulging critical information by RFID tags to any random tags upon interrogation. Engberg *et al* in [33] propose a solution where tags only respond to authenticated readers . The method was developed to avoid customer tracking using RFID tags that the customer may have on them. The method employs the use of a 'zero-knowledge' device authentication method. The method is not technically conventional zero-knowledge, the authors claim zero-knowledge because the tags do not contain any sensitive data. The tags relay the requests to the user/customer upon authenticating a reader to which the customer responds. The solution differs from the extension to DNP3 in that it does not use a challenge/response method to authenticate a sender. The solution involves a user sending a combination of a non-encrypted nonce, and a second nonce using XOR and hash functions. The receiver authenticates the sender on the grounds of knowledge of the shared secret.

The implementation described above employs a shared secret key, but could be extended to use asymmetric methods as well [33]. If the symmetric operation is replaced by an asymmetric operation, the scheme effectively is equivalent to an asymmetric digital signature protocol (e.g. ECDSA presented in the following section).

6.3.4 The TESLA Broadcast Authentication Protocol

The Time Efficient Stream Loss-tolerant Authentication (TESLA) broadcast authentication protocol enables receivers to do source authentication on broadcast messages [10]. The use of symmetric algorithms for authentication fails if the secret key is compromised. Asymmetric cryptographic protocols provide secure authentication but are computationally extensive and have high overhead. The TESLA protocol achieves asymmetric performance while employing purely symmetric cryptographic functions by using delayed key exposure [10]. In the TESLA protocol, the sender attaches to each message a message authentication code (MAC) created with a secret key only known to the sender. The receiver buffers the message since it is unable to authenticate it. The sender at a later time reveals the secret key used to create the MAC, so that the receiver can authenticate the message.

The TESLA protocol has a possibility of a denial of service attack since the messages are buffered until the key is disclosed. An adversary could inject bogus messages into the network and fill up the buffers at the receivers while they are waiting for the key to be disclosed in order to authenticate messages received in a given interval. Such an attack would be to replay a message(s) that is not yet authenticated at the receivers, which would be buffered and cause exhaustion of resources. Until the key for a given interval is exposed, the receivers will buffer all the messages received in a time interval. An attacker could possibly exploit this weakness and cause denial of service.

6.3.5 Summary

The methods described in the previous section offer good authentication solutions but have some drawbacks. The DNP3 solution presented employs a challenge/response method of authentication which cannot be achieved over the RBDS network. The RF fingerprints require additional hardware/DSP for fingerprinting capabilities which could increase the cost of receivers. The Zero-Knowledge device authentication employs symmetric methods which fail if the secret key gets compromised. The TESLA protocol suffers from a denial-of-attack as mentioned previously. The authentication protocols in the next section are identified as candidates for use in residential demand response programs in a one-way communication setting.

6.4 Selected Cryptographical Security Measures

6.4.1 BiBa Signature Protocol

The BiBa protocol as described in [29], is a general solution that can be applied to sign broadcast data based on one-way functions without trapdoors. The BiBa signature scheme is efficient, robust to packet loss and scales well to a large number of receivers. However, the public keys used in the BiBa protocol are large and the time to generate the signatures is long (a BiBa instance with 1024 chains with each value 2 bytes long will have a public key of 2048 bytes). For the purpose of the PCT system, the signature generation overhead can be tolerated. We assume that the sender is equipped with powerful computing resources to handle the signature generation overhead. The small signature sizes make the BiBa protocol a good candidate for the PCT system which is to be deployed over a bandwidth constrained network. A 4-way BiBa instance will have a 8 byte signature (assuming each value in the chain is 2 bytes long). Moreover, the small signature verification overhead allows the end devices (PCT's) to be simple and cheap.

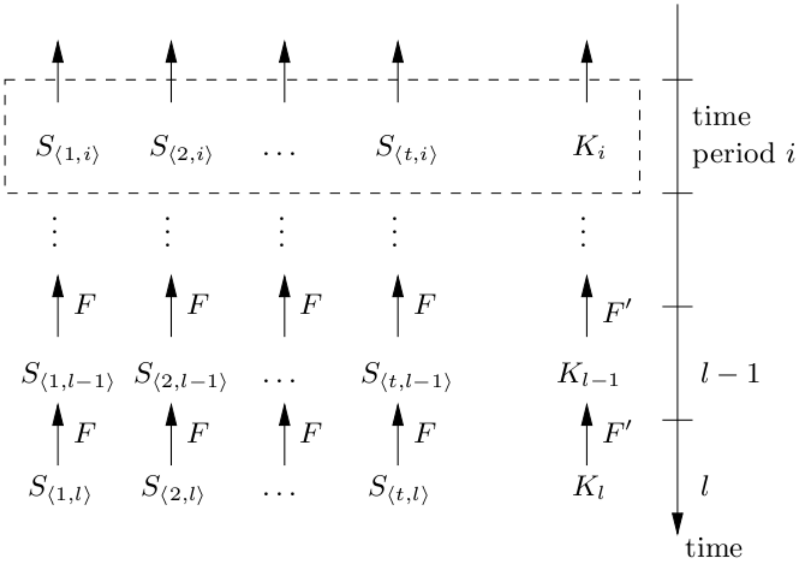


Figure 6.3: The BiBa Broadcast Protocol Dynamics

Figure 6.3 shows the dynamics of the BiBa broadcast protocol. The sender divides time into periods of equal duration. The sender then creates t chains of *SElf Authenticating vaLues* (SEALs), $S_{<1,i>}, ...S_{<t,i>}$, and a Salt chain, K_i, associated with time interval i. The SEAL and Salt chains are of length l, hence they last l time intervals. The Salt key is used by the sender to create the SEALs and is required for authentication of SEALs at the receiver. The SEALs are generated recursively by applying a pseudo-random function F as follows: $S_{<i,j>} = F_{S_{<i,j+1>}}(K_{j+1})$; for $(1 \leq i \leq t)$ and $(1 \leq j \leq l)$. The use of the Salt key forces an attacker to obtain the pre-image of the Salt chain as a pre-requisite to finding the pre-images of the

SEAL chains. Therefore an attacker cannot precompute the SEALs for subsequent time periods without knowledge of the Salt key [29].

At the beginning of each active interval, the sender broadcasts the value of the active Salt (K_i) to the receivers. The dotted box in Figure 6.3 shows an active time interval with the associated Salt key and SEALs. To sign a message m during the active interval i, the sender creates a hash of the message $h = H(m|c)$, which is used to seed a hash function $G_h()$ used to produce a signature; where c is a counter that is incremented when a signature could not be obtained. The sender uses the hash function $G_h()$ on the t SEALs and observes any k-way collisions from distinct SEALs. That is, $S_{<1,i>} \neq S_{<2,i>} \neq .. \neq S_{<k,i>}$ such that $G_h(S_{<1,i>}) = G_h(S_{<2,i>}).. = G_h(S_{<k,i>})$. The k SEALs that result in a collision form the signature and are then sent together with the message as $(< S_1, ..., S_k > ||m)$. The receiver then authenticates the message if $G_h(S_1) = .. = G_h(S_k)$ and $S_1 \neq .. \neq S_k$. During signature generation, it is possible that $G_h()$ applied on all t SEALs fails to produce at least k collisions, in which case a signature cannot be formed. The counter c serves to get a different hash value h in the event that $G_h()$ fails to produce at least k collisions from all t SEALs. The receiver is assumed to know the value k, the hash function H and hash function family G.

The security of the BiBa protocol relies on the fact that a potential attacker knows fewer SEALs than the sender with which to forge a signature. Therefore the sender only reveals the SEALs that are used in creating a signature. The receiver is able to verify that an adversary has a smaller number of SEALs with which to forge a false signature by relying on time synchronization. The BiBa protocol requires loose synchronization between the sender and receiver. When a receiver receives a signed message, it verifies that the sender has not yet revealed r SEALs based on synchronization. If the sender and receivers have a maximum synchronization error of δ, the sender can only send at most $\lfloor r/k \rfloor$ messages within δ time without compromising the security [29]; where r is the maximum number of active SEALs an attacker is allowed to know and k is the number of SEALs revealed in one message. [29] presents a study on how the BiBa protocol can be used in an application and how to determine the BiBa protocol parameters. A receiver is bootstrapped to the sender by revealing all the SEALs and Salt key from one active interval so that subsequent SEALs can be verified. During receiver bootstrapping, the receiver receives the initial values of the SEAL and Salt chains. The receiver then commits to the chains, which allows verification of subsequent SEAL and Salt values. The bootstrap information, which consists of the initial values of the SEAL and Salt salt chains (i.e. the commitment keys of the SEAL chains and the Salt chain), is referred to as the public key in this document. There are extensions that allow efficient bootstrapping of receivers in [29] by periodically sending the SEALs of a time period. The receivers then use the information to verify subsequent SEALs that are used to sign messages.

Authenticating PCT messages using BiBa

The authentication of messages in the PCT system using the BiBa protocol involves a tiered solution. A long-term BiBa instance is used to send short-term BiBa instance commitment keys which serve as public keys. The long-term BiBa instance is conceptually designed to last the entire lifetime of the PCT system. Multiple levels of BiBa instances can be used as necessary to prolong the lifetime of the long-term BiBa instance. Our work uses only two levels to demonstrate the concept and evaluate the performance. Extensions to multiple levels can be done easily following the definition presented here. The long-term BiBa instance is made up of long SEAL chains with large SEAL sizes, hence it is more secure and has a large public key (commitment key). The long-term BiBa instance is used infrequently to bootstrap the receivers to new short-term BiBa instances. The short-term SEAL chains are used to authenticate the application messages using BiBa signatures.

Figure 6.4 shows the dynamics of our authentication construct using the BiBa one-time signature and broadcast protocol. Initially the sender creates a long-term BiBa instance by following the construct described above. The long-term BiBa commitment keys which serve as a public key are then communicated to the receiver(s). The receiver(s) saves the commitment keys to authenticate subsequent messages signed by the long-term BiBa instance. The long-term BiBa instance should be bootstrapped offline or at the time of installation in the case of the PCT system. To allow recovery in the event that the receiver is rebooted, the commitment chain is stored in non-volatile memory. A receiver that is shut down for long periods can synchronize to the short-term BiBa instances by receiving the periodic short-term BiBa instance commitment chains signed by the long-term BiBa instance. The only requirement is that such a device retains the initial commitment key of the long-term BiBa instance.

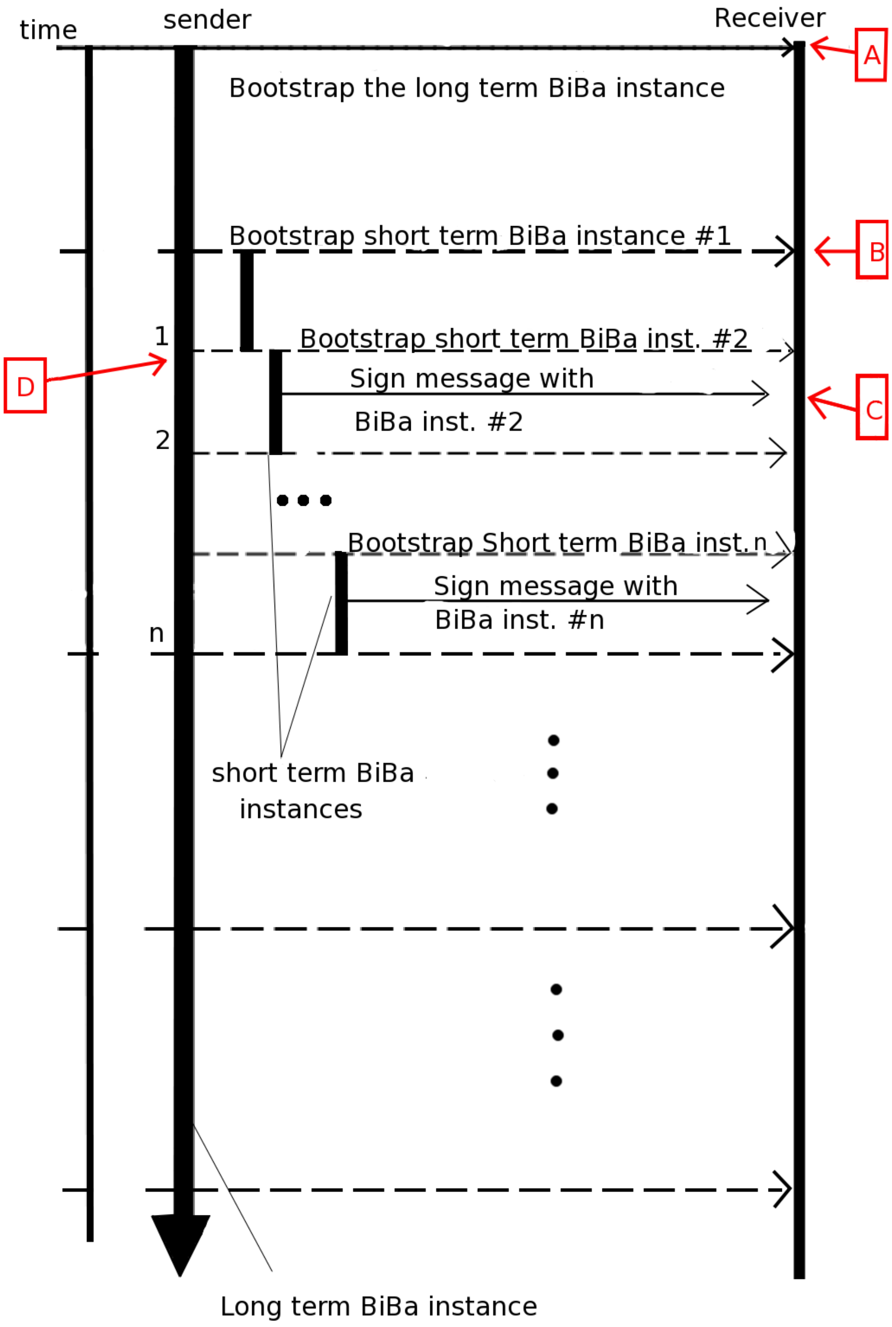

time
sender
Receiver
A
Bootstrap the long term BiBa instance
Bootstrap short term BiBa instance #1
B
1
Bootstrap short term BiBa inst. #2
D
Sign message with
BiBa inst. #2
C
2
Bootstrap Short term BiBa inst. n
Sign message with
BiBa inst. #n
n
short term BiBa
instances
Long term BiBa instance

Figure 6.4: Using the BiBa Signature to Sign Messages

An example illustrating how the protocol works is presented below:

- The receiver commits to the long-term BiBa instance, shown by label A in Figure 6.4. In the PCT system this could be done offline at the time of installation. A technician or home owner keys in the commitment key of the long-term BiBa instance, which is then saved into non-volatile memory on the device. With the long-term BiBa instance commitment keys, the device can authenticate short-term BiBa instance commitment keys signed using the long-term BiBa instance.

- The home device receives the periodic short-term BiBa instance commitment information signed using the long-term BiBa instance, shown by label B in Figure 6.4. The receiver can authenticate the commitment keys of a short-term BiBa instance as described above. When the authentication is successful, the receiver commits to the short-term BiBa instance. The short-term BiBa instance is used to authenticate application messages.

- The home device receives application messages signed using the short-term BiBa instance (shown by label C in Figure 6.4). The application messages are authenticated as described in the definition of the BiBa protocol.

- The short-term BiBa instance expires after l time intervals elapses. Then the long-term BiBa instance creates a new short-term BiBa instance and sends the commitment key to the receivers (illustrated by label D in Figure 6.4).

Protocol Messages

Figure 6.5 shows the structure of the messages sent by the BiBa protocol. A description of the structure of the protocol messages sent to facilitate authentication using BiBa instances is given below. Reference is made to Figure 6.5 to describe the different fields of the messages.

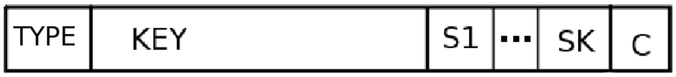


Figure 6.5: Structure of BiBa Messages

TYPE: Describes the type of data that is carried in the message.

0 : Application messages are carried in the KEY field

1 : Short-term BiBa instance commitment key is carried in the KEY field

2 : A Salt key is carried in the KEY field

3 : Long-term BiBa commitment key. This option is not used if the long-term BiBa instance commitment is done off-line

KEY: The Data that is being sent in the message which is signed. Depending on the value of the TYPE field it can either be a message sent by the application or Salt key to be signed by the short-term BiBa instance, or short-term BiBa commitment key.

S1...SK: Part of the signature formed by the k SEALs that resulted in a collision. The size depends on the value of k.

C: The counter that is incremented when a signature is not obtained, which is part of the signature

Security is provided by implementing a security instance in the enabling service layer of the DRI [22]. The security layer interfaces between the application and the physical network. Figure 6.6 shows the actions applied to application messages as they traverse through the different layers. The reverse operation is performed at the receiver. The application generates messages as described in the PCT system. The messages are then delivered to the System Operator who encrypts them for authentication purposes and sends them over the RBDS network. The messages are sent over the RBDS network as type-11A groups. Each RBDS group can only carry 4 bytes of data, so the message is fragmented into multiple RBDS groups and sent over the network. The receiver reconstructs the messages from the multiple RBDS groups and sends it up the protocol stack to the security layer. The security layer then authenticates the messages and present them to the application layer if the authentication is successful.

Setting the BiBa parameters

The parameters of the security layer are based on an approximated application data rate. A BiBa instance with 1024 SEAL chains ($t = 1024$), using 4-way collisions ($k = 4$) can be used to sign 25 messages ($\nu = \lfloor \frac{t\gamma}{k} \rfloor$), with $\gamma = 0.10$; where γ is the fraction of SEALs that can be revealed to an adversary without compromising the

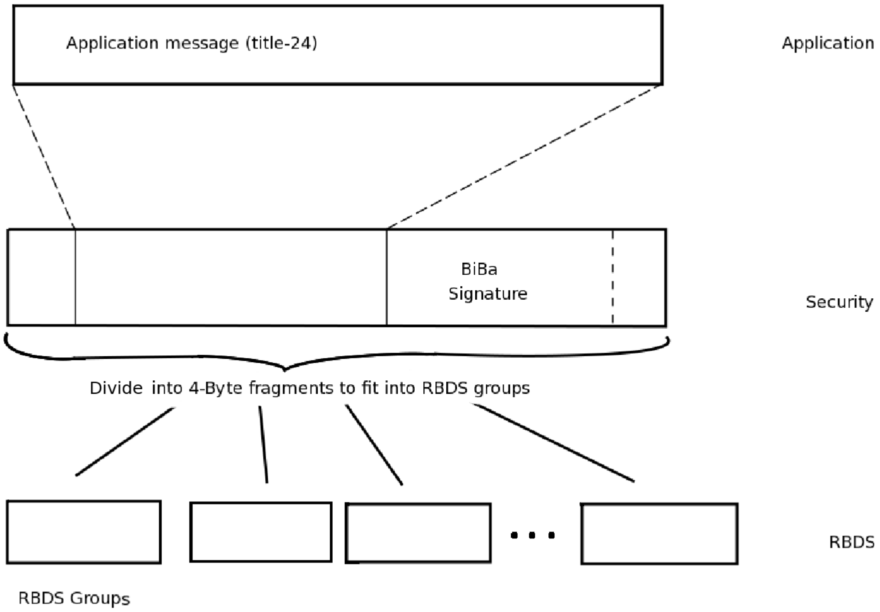

Application message (title-24)
Application
BiBa
Signature
Security
Divide into 4-Byte fragments to fit into RBDS groups
RBDS
RBDS Groups

Figure 6.6: Operations on the Application Messages

security of the protocol (typically $\gamma = 10\%$ [29]). An adversary is only allowed to learn r SEALs from one active period; where $r = t\gamma$. Each signature reveals k SEALs to the adversary, hence only ($\nu = \lfloor \frac{t\gamma}{k} \rfloor$) messages can be signed within a single time interval. An adversary who knows r SEALs needs to make 2^{35} computations to forge a valid signature of a BiBa instance with the above parameters according to [29]. If we assume that the application sends an average of 20 event messages every day, a single time interval for the short-term BiBa instance is sufficient to authenticate an entire day's messages. Consequently, a short-term BiBa instance with SEAL chain lengths of 50 ($l = 50$), can be used for 50 days before it expires. If the long-term BiBa instance is designed with the same parameters as the short-term instances (i.e. $t = 1024$, $k = 4$, $\gamma = 10\%$), then it can be used to commit 25 short-term BiBa instances in a single time interval. A single time interval for the long-term BiBa instance can then be made to last up to 1250 days (3.4 years). The entire long-term BiBa instance will then last 171 years.

6.4.2 HORSE Authentication Protocol

The HORSE authentication protocol extends the HORS (Hash to Obtain Random Subsets) protocol which is an extension of the BiBa protocol to provide broadcast authentication. HORSE and BiBa are r-time signature schemes that provide unforgeable signatures which can be verified by using publicly available information. R-time signature schemes achieve faster signature generation at the expense of larger key sizes. Generally, the generation of such signatures is faster than public-key signatures but they can only be used to sign r messages [38].

The HORS protocol works by mapping a message m to a k-element subset of t-element set T. The mapping of a message m is achieved by a collision-resistant hash function H (eg. MD5 or SHA-1). Then, for messages m_1 and m_2; $m_1 \neq m_2$, it should be impossible to get $H(m_1) \subseteq H(m_2)$. In a general case for r messages $m_1, m_2, ..., m_r$, it must be infeasible to obtain $H(m_r) \subseteq \bigcup_{i=1}^{r-1} H(m_i)$. To obtain the k-element subset, the output of the hash function $H(m)$ is split into k substrings each $\log_2(t)$ bits. The substrings are then interpreted as integers j_i, $(1 \leq i \leq k)$, which selects k values in set T. The k values selected from T form the signature $(s_{j_1}, s_{j_2}, ..., s_{j_k})$.

To sign a message m in HORS, the sender initially selects values t and k such that $k \log_2 t \leq |H(\cdot)|^2$. The function H as described above is a collision resistant hash function that maps a message m to k-element subsets of T. The sender then generates the secret key, SK $= (s_1, s_2, ..., s_t)$ by randomly generating t l-bit values. The public key is then, PK $= (v_1, v_2, ..., v_t)$ with $v_i = f(s_i)$,$1 \leq i \leq t$,where f is a one way function. The sender computes $h = H(m)$, and splits h into k sub-strings each of length $log_2 t$ bits. Each sub-string is interpreted as an integer j_i for $(1 \leq i \leq k)$. The signature is then made of the subset of SK, $(s_{j_1}, s_{j_2}, ..., s_{j_k})$ and is sent along with the message m. To verify a signature $(s'_1, s'_2, ..., s'_k)$ at the receiver, the receiver computes $h = H(m)$. The receiver then splits h into k substrings and interprets the substrings as integers of $log_2 t$ bits. Then it verifies that $v_i = f(s'_i)$, for $1 \leq i \leq k$ otherwise the signature is rejected.

HORSE extends HORS by using one way chains to generate and update the secret key and public key pair. In the HORS protocol one can only sign r messages without losing security. HORSE uses a one way hash function $H()$ to generate chains of values each d values long. To initialize, the sender generates t random values $(s_{<0,1>}, s_{<0,2>}, ..., s_{<0,t>})$ and uses them to construct t chains of length d. The hash function is used recursively d times on each of the t initial values to get a chain as shown in Figure 6.7. The keys are then used in reverse order of generation. That is, the initial secret key is given by $SK_0 = (s_1, s_2, ..., s_t) = (s_{<d-1,1>}, s_{<d-1,2>}, ..., s_{<d-1,t>})$, where $s_{<i,j>} = H^i(s_{<0,j>})$. The initial public key is given by $PK_0 = (v_1, v_2, ..., v_t)$; $v_i = f(s_i)$, $\forall s_i \in SK_0$

The signature generation and verification is computed as described above for HORS. The secret-key gets updated after each signature is generated. The values

$$\begin{array}{|cccccc|c|}
\hline
s_{\langle 0,1\rangle} & s_{\langle 0,2\rangle} & \cdots & s_{\langle 0,t-1\rangle} & s_{\langle 0,t\rangle} & & \\
\downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & \cdots & \downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & & \\
s_{\langle 1,1\rangle} & s_{\langle 1,2\rangle} & \cdots & s_{\langle 1,t-1\rangle} & s_{\langle 1,t\rangle} & & \\
\downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & \cdots & \downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
\hline
s_{\langle d-1,1\rangle} & s_{\langle d-1,2\rangle} & \cdots & s_{\langle d-1,t-1\rangle} & s_{\langle d-1,t\rangle} & & SK_0 \\
\hline
\downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & \cdots & \downarrow \mathcal{H}(\cdot) & \downarrow \mathcal{H}(\cdot) & & \\
\hline
s_{\langle d,1\rangle} & s_{\langle d,2\rangle} & \cdots & s_{\langle d,t-1\rangle} & s_{\langle d,t\rangle} & & PK_0 \\
\hline
\end{array}$$

Figure 6.7: The HORSE Protocol

used to generate the signature gets replaced by the values preceding them in the respective chains as shown by Figure 6.8. The Figure depicts a scenario where the secret key SK_i gets updated after using the values $s_{<\mu,\alpha>}$, $s_{<\nu,\beta>}$, and $s_{<\xi,\kappa>}$ in a signature. The secret key is then updated to SK_{i-1} with $s_{<\mu,\alpha>}$, $s_{<\nu,\beta>}$, and $s_{<\xi,\kappa>}$ replaced by $s_{<\mu-1,\alpha>}$, $s_{<\nu-1,\beta>}$, and $s_{<\xi-1,\kappa>}$ respectively, while the other values remain unchanged.

$$\begin{array}{ll}
SK_i = & (s_{\langle\zeta,1\rangle}, s_{\langle\eta,2\rangle}, \ldots, \; s_{\langle\mu,\alpha\rangle}, \; \ldots, \; s_{\langle\nu,\beta\rangle}, \; \ldots, \; s_{\langle\xi,\kappa\rangle}, \; \ldots, s_{\langle\rho,t\rangle}) \\
 & \qquad\qquad\quad \downarrow \mathcal{H}^{-1}(\cdot) \quad \downarrow \mathcal{H}^{-1}(\cdot) \quad \downarrow \mathcal{H}^{-1}(\cdot) \\
SK_{i+1} = & (s_{\langle\zeta,1\rangle}, s_{\langle\eta,2\rangle}, \ldots, s_{\langle\mu-1,\alpha\rangle}, \ldots, s_{\langle\nu-1,\beta\rangle}, \ldots, s_{\langle\xi-1,\kappa\rangle}, \ldots, s_{\langle\rho,t\rangle})
\end{array}$$

Figure 6.8: Updating the Public and Secret Keys in HORSE

The receiver updates the public key each time it receives a signed message. The receiver verifies the signature by performing a hash operation on the values that make up the signature and compares them to the public key as described earlier. After successful verification, the receiver updates the public key by replacing the values in the public key that are preceded by the values that make up the received signature. In the example shown in Figure 6.8, the values that make up the signature ($s_{<\mu,\alpha>}$, $s_{<\nu,\beta>}$,$s_{<\xi,\kappa>}$) replace the corresponding values in the public key (v_α, v_β,v_κ) in the public key. In a lossy environment, the receiver may not successfully receive the signed message and lose synchronization, which would lead to unsuccessful signature verification. To avoid the loss of synchronization, the index corresponding to the position of the values that make up the signature in their corresponding chains is sent as part of the signature. This lets the receiver know how many hash operations it needs to perform to verify each value in the signature. The signature is then formed by ($< \alpha_1, s_1 >$, $< \alpha_2, s_k >$,$< \alpha_k, s_k >$) where $\alpha_i \in [0, d-1]$ gives the position of s_i in the chain. The signature is then verified if $H^{d-\alpha_i}(s_i) = s_{<d,i>}$ for $1 \leq i \leq k$.

In the worst case, the maximum number of messages that can be signed is d. That would happen if at least one chain gets used to create a signature every time a message is sent. Based on probability, the expected number of messages that can be signed is $d/(1-e^{-k/t})$ [38]. As an example, in [38] HORSE is expected to sign up to $65 \cdot d$ messages compared to 4 messages for HORS, for the same parameters $t = 1024$ and $k = 16$. The tradeoff is that the memory required to store the chain values in HORSE is d times that of HORS. Alternatively, if memory is not enough, HORSE could require up to $k \cdot d$ hash evaluations to generate each signature. However, [38] mentions a technique that allows efficient storage of chain values that requires only storing $\log_2 d$ hash values and performs at most $\log_2 d$ hash evaluations per step.

Authenticating PCT Messages Using HORSE

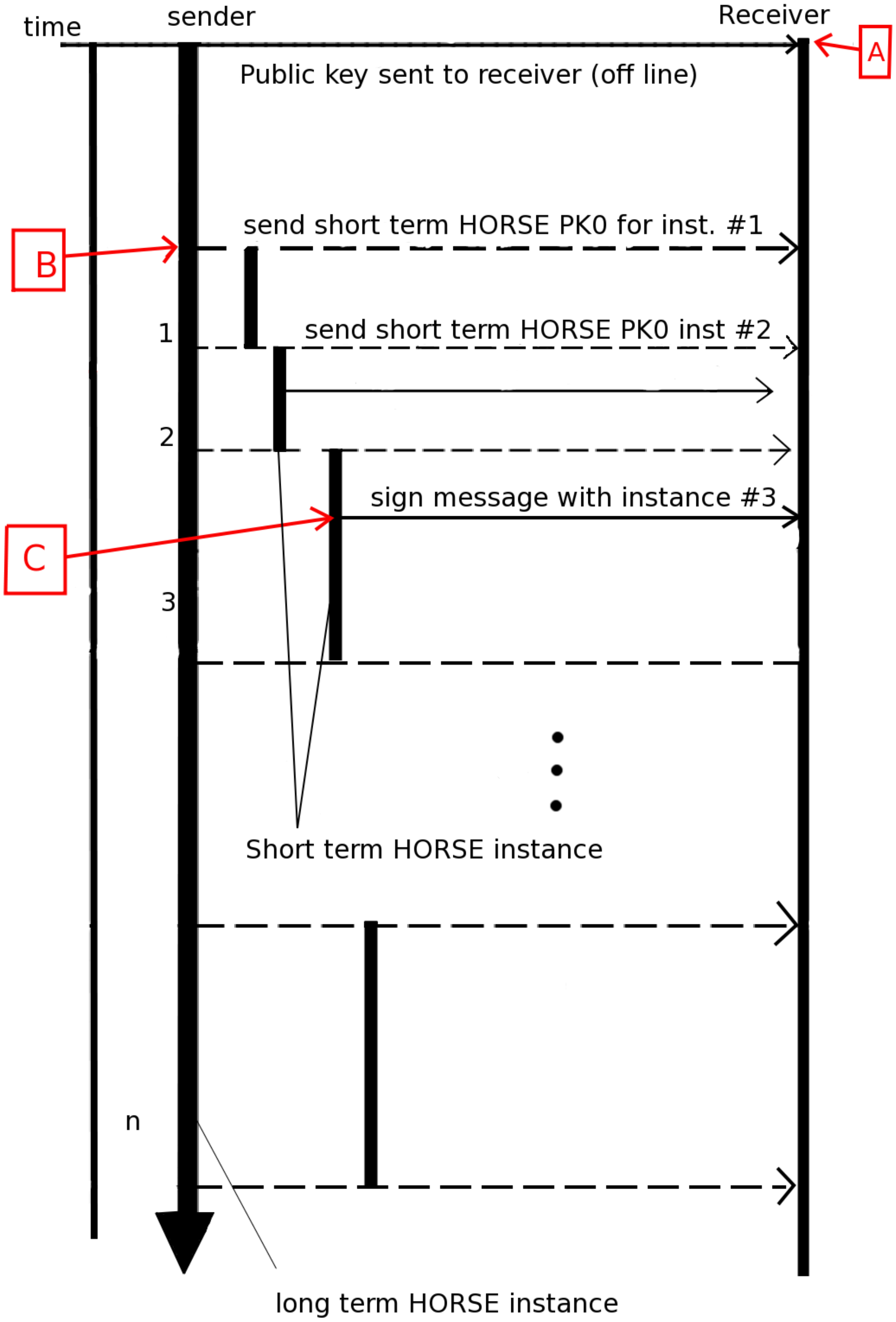

time
sender
Receiver
A
Public key sent to receiver (off line)
send short term HORSE PK0 for inst. #1
B
1
send short term HORSE PK0 inst #2
2
sign message with instance #3
C
3
Short term HORSE instance
n
long term HORSE instance

Figure 6.9: Employing HORSE to Authenticate Messages

Each HORSE instance is expected to sign $n = d/(1 - e^{-k/t})$ messages on average as explained above. To sign messages exceeding $d/(1 - e^{-k/t})$, there is a need to use a new HORSE instance after the current one expires. To address the issue of signing messages exceeding $d/(1-e^{-k/t})$ messages, we propose a tiered solution similar to the one employed for using BiBa to sign PCT messages. A long-term HORSE instance is used to send the initial public key of a new short-term HORSE instance when the current one expires. The initial public keys of the short-term HORSE instance are signed by the long-term HORSE instance and sent to the receivers to allow the receivers to verify subsequent messages. Figure 6.9 shows the structure of the construct to provide authentication for the PCT system.

An example to illustrate how the HORSE construct works is presented below:

- The initial public key of the long-term HORSE instance is sent to the receivers at the time of installation. This is done by a technician installing the PCTs, shown in Figure 6.9 by label A.

- Short-term HORSE initial public keys are then sent to the receivers signed using the long-term HORSE private key as described above. Label B in Figure 6.9 shows a short-term initial public key being sent to the receivers.

- Application messages are signed with the short-term HORSE instance as described previously, shown by label C in Figure 6.9.

- Each short-term instance HORSE on average will send $d/(1 - e^{-k/t})$ messages after which a new short-term HORSE instance needs to be created. When one of the chains in the short-term HORSE instance is about to be exhausted (left with say 3 values), a new short-term HORSE instance is created and the public key is sent to the receivers. When the first chain is exhausted (left with 1 value), a message is sent to the receivers to instruct them to use the last public key they received. The message sent to the receiver to switch to the new public key is encrypted with the expiring short-term HORSE instance, such that the chain that had 1 value left have all its values used.

A short-term HORSE with $t = 1024$, $k = 4$, and $d = 50$ on average will sign 12825 messages before a new short-term instance is required. If the long-term HORSE instance has the same parameters, then 12825 short-term HORSE instances can be signed. Therefore on average $12825^2 = 164480625$ application messages can be signed. Keeping the previous application data rate of 20 messages per day, the construct can last 8224031 days (22531 years).

HORSE Protocol Messages

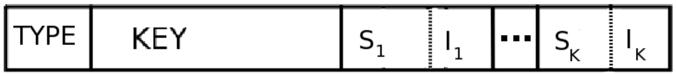


Figure 6.10: Structure of HORSE Protocol Messages

The protocol messages that are communicated to facilitate the use of HORSE for authentication in our solution follow the format used for BiBa (See Section 6.4.1). The structure of the messages is shown in Figure 6.10. The message fields are defined as follows:

TYPE: Describes the type of data that is carried in the message.

0 : Application messages are carried in the KEY field

1 : Short-term HORSE instance public key is carried in the KEY field

2 : Command to switch to newly received public key

3 : Long-term HORSE commitment key. This option is not used if the long-term HORSE instance commitment is done off-line

KEY: The Data that is being sent in the message which is signed. Depending on the value of the TYPE field it can either be a message sent by the application or a command to switch to the next HORSE instance, or short-term HORSE public key.

$< S_1, I_1 >$**...**$< S_K, I_k >$**:** The signature formed by the k values that forms the subset (S_i), and the index of the values in the chains(I_i); with $1 \leq i \leq k$. The size depends on the value of k.

The flow of messages will follow the same steps as depicted in Figure 6.6

6.4.3 The Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm (ECDSA) is similar to the Digital Signature Algorithm (DSA), but employs elliptic curves over a finite field[16]. Elliptic curve cryptography offers faster verification and smaller keys for equivalent security with other public key systems [11]. Based on the complexity of the Elliptic Curve Discrete Logarithmic Problem, it is computationally infeasible to forge a signature if appropriate parameters are employed.

A finite field F is made up of a finite number of elements together with two binary operations on F. The binary operations, addition and multiplication have special arithmetic properties as defined in [16]. The order of a finite field is the number of elements in the field. If p is a prime number, then the field F_p is called a prime field and is made up of integers $\{0,1,2,...,p-1\}$. Addition and multiplication of elements of F_p are done modulo p. That is $a+b=r$; where $r=(a+b)$ **mod** p, and $a \cdot b = s$; where $s = a \cdot b$ **mod** p. An elliptic curve E on a finite field F_p; where $p>3$ is an odd prime, is given by the equation:

$$y^2 = x^3 + ax^2 + b \tag{6.1}$$

where p is a prime number, $a, b \in F_p$, and $(4a^3 + 27b^2)(\mathbf{mod}\ p) \neq 0$. The set $E(F_p)$ consists of all points (x, y) $(x, y \in F_p)$ that satisfy equation 6.1 and a point ϑ located at infinity. The point ϑ is the identity element of the group $E(F_p)$.

All the elements of the set $E(F_p)$ have the properties:

$$P + (-P) = (-P) + P = \vartheta$$

and

$$P + \vartheta = \vartheta + P = P$$

for $P \in E(F_p)$

The security of elliptic curve cryptography comes from the Elliptic Curve Discrete Logarithmic Problem (ECDLP). The ECDLP consists of finding a value k such that $P = kQ$ given P and Q (with $P, Q \in F_p$). There is no efficient known algorithm that can compute the value of k [11]. The parameter requirements to achieve resilience to known attacks are outlined in [16]. [16] also gives ways of generating cryptographically secure parameters for elliptic curves using several methods.

To sign a message, initially the sender and receiver agree on an elliptic curve with a base point P over the field F_p. The sender has a private key x and a public key $Q = xP$. The parameters of the curve a, b, P, q, F_p as well as the public key Q are assumed known to the receiver. To sign a message m:

1. The sender generates a random number k; $k \in [1, n-1]$ and then computes $kP = (x_1, y_1)$. The value x_1 is then converted to an integer $\bar{x_1}$
2. Compute $r = \bar{x_1}\textbf{mod } n$. If $r = 0$ then step 1 is repeated until $r \neq 0$.
3. Compute $k^{-1}\textbf{mod } n$.
4. Compute SHA-1(m) and convert the output string into an integer e.
5. Compute $s = k^{-1}(e + dr)\textbf{mod } n$; with $s \neq 0$. If $s = 0$ then go back to step 1.
6. The pair (r,s) forms the signature and is sent to the receiver.

When the receiver receives the signed message it performs the following steps to verify the signature.

1. Confirm that $r, s \in [1, n-1]$.
2. Compute SHA-1(m) and convert the output string into an integer e.
3. Compute $w = s^{-1}\textbf{mod } n$.

4. Compute $u_1 = ew\mathbf{mod}\ n$ and $u_2 = rw\mathbf{mod}\ n$.

5. Compute $X = u_1P + u_2Q$. If $X = \vartheta$ reject the signature.

6. Convert the x coordinate of X to an integer $\bar{x_1'}$, and compute $v = \bar{x_1'}\mathbf{mod}\ n$

7. Accept the signature if $v = r$

Authenticating PCT Messages Using ECDSA

Authenticating PCT messages does not require a lot of changes to ECDSA. The use of the hash function SHA-1 in step 2 of the signature generation can be omitted since the PCT messages are small. Step 2 at the receiver will also be omitted, resulting in less computations. The use of the hash function is to obtain a fixed length string from variable message lengths. In practice the messages could be a file a few kilo bytes long, hashing it using SHA-1 results in a string 160 bits long.

The value $n > 2^{160}$ is recommended to protect against attacks as outlined by [16]. The sizes of the signature (r,s) is dependent on the value n, $r, s \in [1, n-1]$. Taking a minimalist approach and using $n = 2^{160}$, we have both r and s being 20 bytes long (the signature is 40 bytes long). The above parameters can then be used for the PCT system for authentication purposes. It may be necessary to use a dynamic approach where the elliptic curve parameters used are refreshed periodically. To allow such a construct, the sender uses one long-term secret key to sign the new parameters when sending to them to the receivers.

6.5 Simulation Results and Analysis

Security is provided by implementing the above three authentication schemes in the enabling services layer of the DRI [22]. Figure 6.11 shows the resultant communication protocol stack. The application sends messages down the stack to the security layer. The security layer signs the messages and sends the signed message over the RBDS network. RBDS is dependent upon to fragment the signed message into RBDS groups at the sender side and reconstruct the signed message from fragments at the receiver. The security layer at the receiver end receives the signed message from the RBDS and decyphers it. Upon successful verification of the signature, the message is sent up to the application, running on the end device.

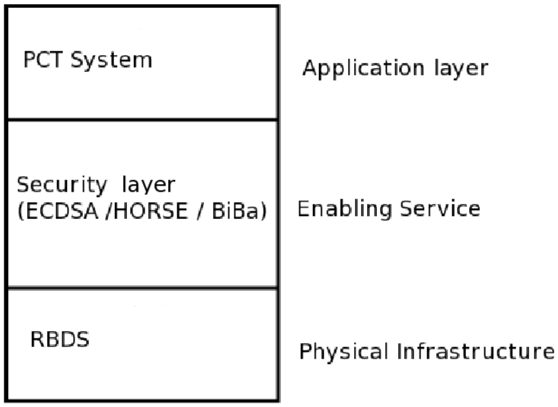


Figure 6.11: Communication Protocol Stack with Security Features

Table 6.1 shows the parameters of the RBDS network used to evaluate the security protocol. All results presented here, unless stated otherwise are based on the parameters in Table 6.1. A typical dispatch message used for curtailing loads is expected to 20 Bytes according to [1]. In our simulations we use messages 30 Bytes to account for additional addressing overhead. We do not expect our proposed addressing scheme to exceed 10 Bytes in length even in the extended mode of addressing.

RBDS Mode	Real-Time
Number of re-transmissions	5
Raw data bitrate	1187.5 bps
Propagation model pathloss exponent	3.50
Transmission power	27kW
Application Message size	30 Bytes

Table 6.1: Physical Network Parameters

6.5.1 BiBa Performance Results

From the simulations of the security protocol, a BiBa instance of 1024 SEAL chains was found to take 440 seconds (7.33 minutes) to bootstrap, assuming that no other application uses the network. Reducing the number of SEAL chains by half reduces the bootstrapping time by half. Alternatively, SEALs of smaller size could be used to reduce the size of the public key. The time taken to bootstrap a new BiBa instance needs to be short to avoid periods where the receiver is not synchronized with the sender. If such periods are allowed, the receiver will be unable to authenticate the new messages. To avoid such a problem, the last few time intervals (depending on the size of the short-term BiBa commitment key) of the current short-term BiBa instance could be used to bootstrap the next short-term BiBa instance.

The commitment keys used to bootstrap BiBa instances (public keys) are in the order of a few kilo bytes. A BiBa instance with 512 value chains with each value 2 bytes long will have a public key of 1 KB. Successful reception of such messages over the lossy RBDS channel becomes a problem as shown in Section 4.2. Initial studies of the RBDS network show that the probability of receiving messages sent over the network is inversely proportional to the size of the message. The size of the BiBa commitment key using 512 SEAL chains ($t = 512$), with each SEAL 16-bits (2 Bytes) long for the short-term BiBa instance, the key is $m = \frac{512*2}{RBDS\_Group\_size} = 256$ RBDS groups; with ($RBDS\_Group\_size = 4$). Such a large value for m diminishes the probability of reception rapidly as shown by Figure 4.5.

The trick mentioned earlier of avoiding unsynchronized periods between successive BiBa instances helps to increase the probability of receiving a new public key to an already synchronized node. This is because such a node effectively has 2 chances of receiving a commitment key for the next short-term BiBa instance. The first chance comes from using the long-term BiBa instance to bootstrap the new short-term BiBa instance. The second chance comes from using the short-term BiBa instance to avoid unsynchronized periods between successive short-term BiBa instances. To increase the chances of bootstrapping short-term BiBa instances the number of times a public key is sent can be increased by sending commitment keys by both the long-term and short-term BiBa instances.

Different values for the number of SEALs in the short-term BiBa instance (t) have an impact on the reception of application messages. Figure 6.12 shows how the probability of receiving application messages vary with different commitment key sizes with 95% confidence intervals shown by the error bars. The results presented in Figure 6.12 are for fixed SEAL sizes (16 bits). A larger value for t results in a larger commitment key which has lower probability of reception. It can be seen that generally a larger value of t results in lower reception probability compared to smaller values of t. It can be seen in the figure that as the distance increases the 95% confidence interval of the estimated reception probability increases. At distances beyond 120 km the variation confidence interval increases, at 140 km the interval is ±5% and increases to ±7% at 180 km. This is due to the vast variation of reception of messages depending on the bootstrapping of devices and increased message sizes. It is worth noting that at distances beyong 120 km the service availability of 95% cannot be achieved hence one should not operate at those numbers.

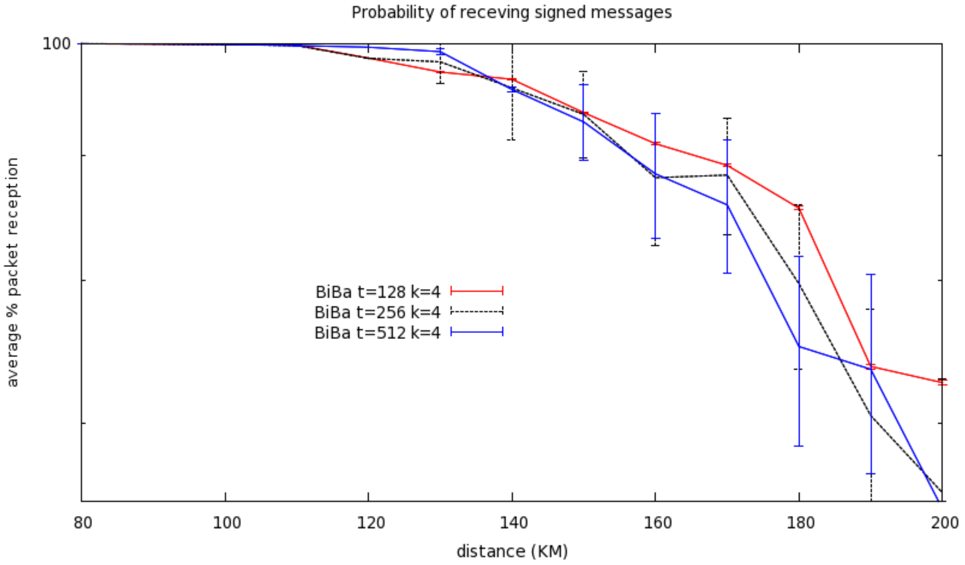


Figure 6.12: The Effects of Public Key Sizes on the Reception of Messages

The reception of application messages depends largely on the successful bootstrapping of the BiBa instances at the receiver. When the receiver fails to bootstrap, all the messages received during that period will not be successfully verified. The results depicted in Figure 6.13 show degraded performance against the case with no security because of the obvious bootstrapping problem caused by large commitment keys. More application messages are rejected by the security layer caused by unsynchronised receivers when bootstrap information is not received. The successful reception of messages varies accordingly with the number of retransmissions of RBDS groups. By increasing the number of RBDS re-transmissions, the reception of the individual messages and more importantly the commitment keys, will be increased.

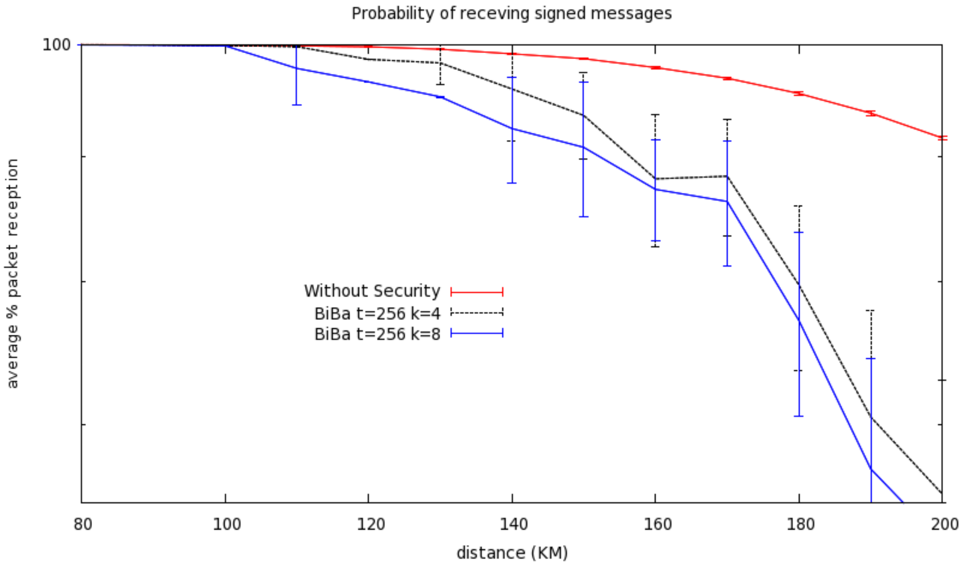


Figure 6.13: Effect of Signature Sizes on Message Reception

The signature sizes are linear in k for a k-way BiBa signature, and public keys are linear in the number of SEALs (t). The signed messages are bigger than unsigned messages because of the signatures. Increasing the number of collisions required for the signature (k) increases the sizes of the transmitted message but also increases the security of the BiBa protocol by lowering the chances of an adversary to successfully forge a signature. It is expected that messages bearing larger signatures incur lower reception at the receiver based on initial studies of the RBDS network. Figure 6.13 shows the probability of receiving messages using different values of k. The graph on top shows the probability of reception of messages with no authentication. As shown by Figure 6.13, the messages of an 8-way BiBa signature scheme are less likely to be received than those of a 4-way BiBa scheme. The difference between the signed messages and unauthenticated messages is large because of the dependence on successful bootstrapping of receivers for signed messages. The successful bootstrapping of receivers diminishes fast with distance and effectively reduces the coverage area that a Systems Operator can offer high availability of services.

The signatures introduce significant overhead to messages if the messages are small. An 8-way BiBa signature scheme with 16-bit SEALs has a 100% overhead on 16-Byte messages without considering the bootstrap keys. The messages that are transmitted in the PCT system are very small messages, in the order of tens of bytes. Therefore the signature overhead is large and together with the communication of public keys, the BiBa protocol as used in our design is not bandwidth efficient. However, if the SEAL chains are long and each BiBa instance lasts for long periods of time, the consumption of bandwidth for security reasons can be very small. If one commitment key is sent every 24 hours to bootstrap the receivers, then the bandwidth consumption by background traffic is small. For a short-term BiBa instance with 512 SEAL chains, each SEAL 2 Bytes long, the bandwidth consumption will be $\frac{512*2*8}{24*60*60} = 0.0948bps$. With this construct, the availability of the service would be increased by multiple transmissions of the commitment keys for the short-term BiBa instance.

The operation of the RBDS network calls for careful design of the interaction of the security layer and other applications running on top of the RBDS network. From simulations there are instances when the application messages can pre-empt the transmission of messages when RBDS is operating in Real-Time Mode. If such an event occurs while the short-term BiBa commitment keys are being transmitted, the security protocol would perform badly. Therefore, both the application using the security protocol and other applications running on top of RBDS need to be designed to avoid pre-empting the security commitment keys.

6.5.2 HORSE Performance Results

Simulation conditions for HORSE were kept equivalent for the case employing BiBa as the security protocol. To be specific, the memory required to store the one way chains at the sender was kept constant. Therefore, the number of chains, chain value sizes and chain lengths were kept equal for both cases. Fixing the mentioned parameters results in an equivalent size of the public keys for HORSE and BiBa. To improve the chances of successful bootstrapping to new HORSE instances, the initial public keys could be sent periodically, the same as described for BiBa.

The size of the signature in HORSE is larger than in the case where BiBa is employed. The increase in signature size is a result of including the position of the values that make up the signature in their respective chains. The signature size in bits is $\lceil log_2 d \rceil \cdot k$. Therefore, the signature size increases linearly with the number of values that make up the signature,(k), and logarithmically with the lengths of the chains used to generate the signatures (d). There is a tradeoff between the lengths of the chains d, and the storage requirements and signature size. A large value of d results in a higher number of messages that can be signed by a single HORSE instance. On the other hand, a large d requires more storage and/or computation by the sender and larger signatures. The computational overhead at the sender for the PCT system can be tolerated since the base station is assumed to be equipped with powerful storage and computing resources. Figure 6.14 shows the relation between different values of k. As depicted by Figure 6.14, unsigned messages (no signature or $k = 0$) have a better probability of reception than the signed messages. The messages with larger value for k have lower chances of reception as shown by the curve for $k = 8$ against that of $k = 4$. The reduction in performance is a result of the signatures, which result in large messages being sent over the RBDS network. The difference between the signed messages and the unsigned messages is large due to the dependence on successful bootstrapping of the devices.

The results presented in Figure 6.14 and the rest of the document are for a fixed value of d, where an 8-bit unsigned value carries the index of the values making up the signature. An 8-bit value for the index can represent positions of values for chains up to $d = 256$ values long. It is not expected to have chains more than 256 values in length because that could potentially result in extensive computations at the receivers to verify signatures. To verify a signature, the receiver performs up to $d \cdot k$ hash operations. Large values of d would place a lot of computational burden on the PCT receivers which are expected to have low computational power.

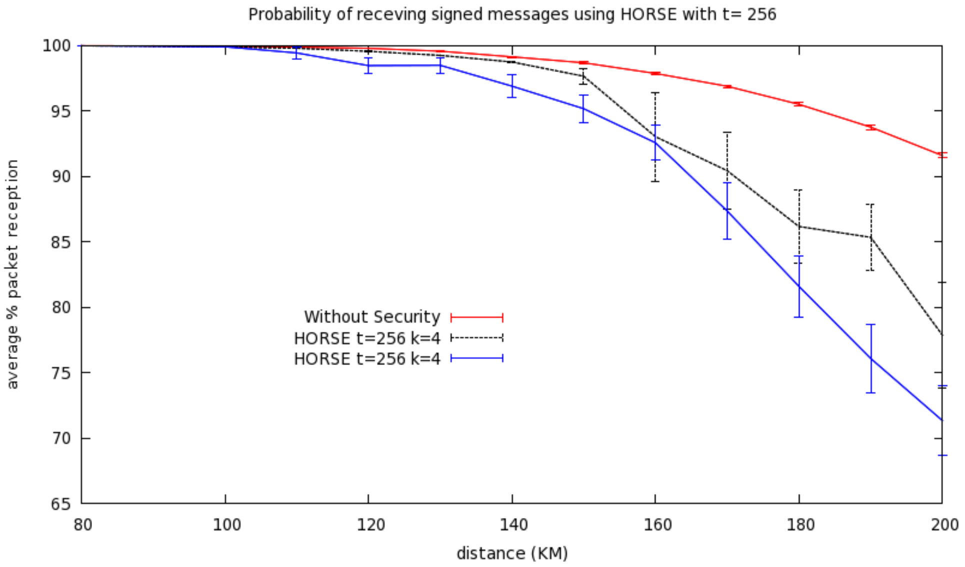


Figure 6.14: Effect of Different Signature Sizes on Message Reception

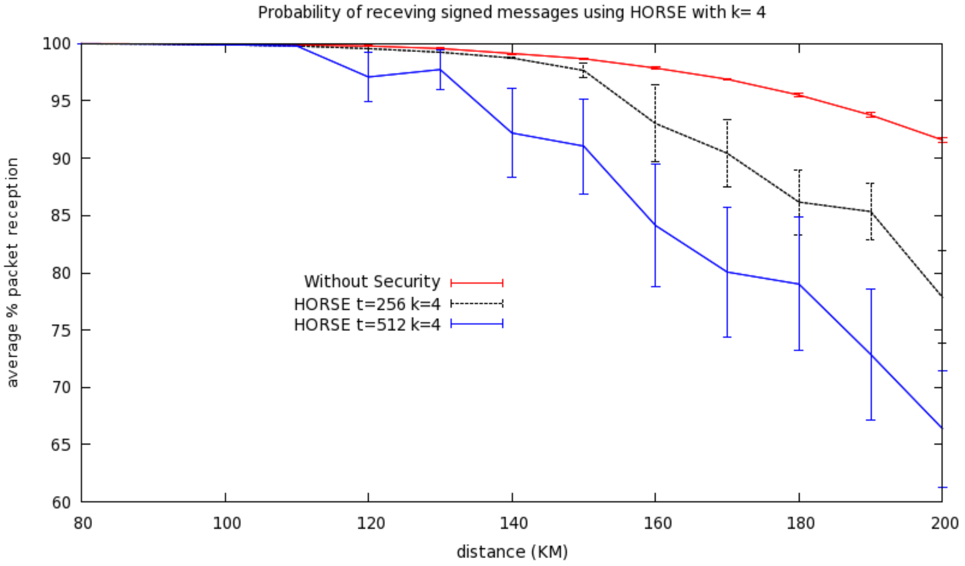


Figure 6.15: Effect of Public Key Sizes on Message Reception

Successful reception of initial public keys is critical for the verification of signatures. The larger the signatures, the more likely that the initial public key is not received successfully. In this case, the receiver will not be synchronized with the sender and will fail to verify signed messages. Figure 6.15 shows the effect of varying values of t. The size of the public key grows linear with respect to t. To keep the cases with different values of t equivalent, the value $t \cdot d$ was kept constant in all cases. That is, the amount of storage required by the sender is the same in all cases. The results are as expected with larger values of t performing worse than smaller values. This is shown in Figure 6.15 with $t = 256$ performing better than $t = 512$.

6.5.3 ECDSA Performance Results

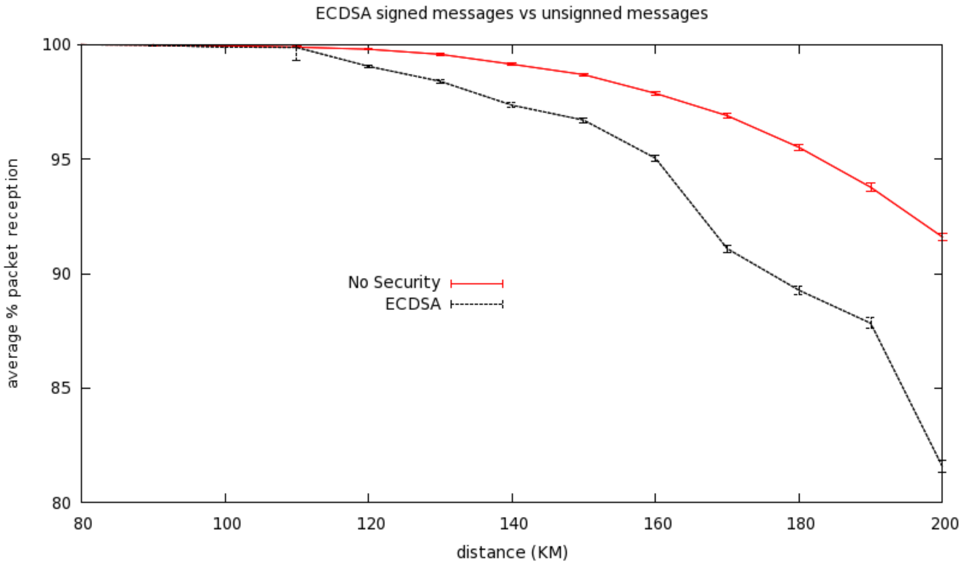


Figure 6.16: Performance of ECDSA Used Over the RBDS Network

The ECDSA protocol produces large signatures compared to BiBa and HORSE. For the recommended 160-bit key (according to [16]) ECDSA has a signature of 40 bytes. This introduces a lot of overhead in the case of PCT messages which are expected to be a few tens of bytes in size. The large messages result in lowered performance as shown by Figure 6.16. It is worth noting that although the overhead on each message is large for ECDSA, the successful verification of signatures does not depend on receiver bootstrap as is the case with BiBa and HORSE.

6.5.4 Comparing the Authentication Protocols

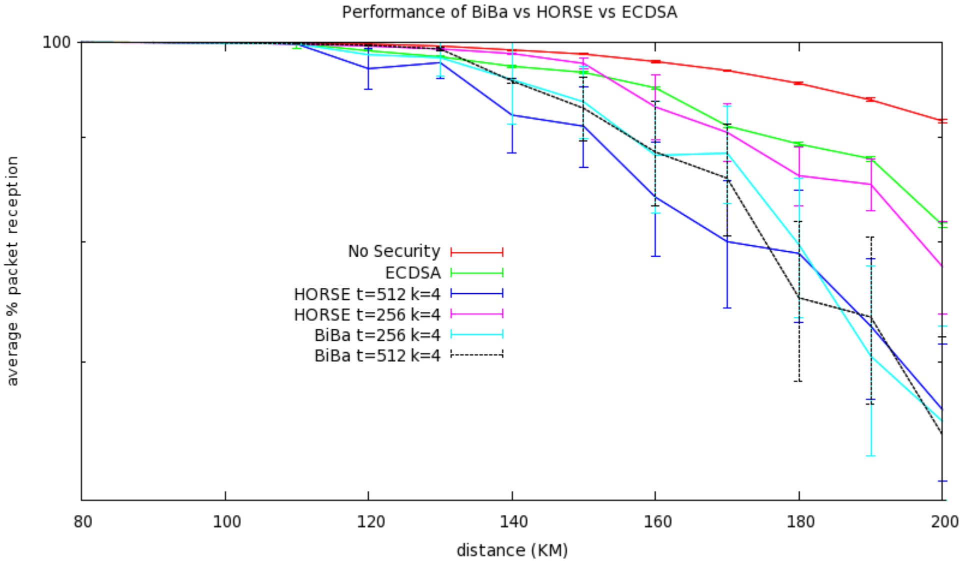


Figure 6.17: Comparisons Between BiBa, HORSE and ECDSA

Figure 6.17 puts things in perspective and compares the performance of all the protocols against each other. Figure 6.17 shows the reception of messages signed by each of the authentication protocols under investigation. From the simulation results as shown in Figure 6.17, HORSE performs better than BiBa in terms of successful reception of messages. For the same values of k and t, HORSE has a significant advantage over BiBa as shown in the figure. As an example, at a distance of 140km, HORSE with $t = 256$ and $k = 4$ gives 5% better message reception than BiBa with the same parameters. The difference in performance increases as the distance increases, reaching 20% at a distance of 180km. The performance improvement in HORSE comes from a more efficient use of chain values. In BiBa, the short-term BiBa instance expires when the time set for it elapses, even if there are no application messages sent during that period. This results in frequent transmission of the large commitment keys (public keys) which have low chances of successful reception. The dependence on successful bootstrapping of the devices account for the big fluctuations in graphs for both BiBa and HORSE. As it can be seen by the increasing error margin as distance increases in Figure 6.17. This is due to the fact that when a a device loses synchronization all signed messages cannot be successfully verified and hence do not reach the application. From the figure, there is no significant difference between HORSE with $t = 256$ and $k = 4$ and ECDSA. It is worth noting that with HORSE the values of t and k can be adjusted to improve performance, albeit with tradeoffs mentioned previously.

ECDSA has large signatures on each message and had overhead of $\frac{40}{30} * 100\% = 133.33\%$ on a 30 byte message (the signature is 40 bytes long). A HORSE instance with parameters $t = 256$, $k = 4$, and $d = 20$ introduces only 41.32% overhead. Such a HORSE instance can be used to sign $d/(1 - e^{-k/t}) = 1290$ messages. Each message is 30 bytes long and the signature on each message is $k(b + a) = 12$ bytes long(where $b = 2$ and $a = 1$; i.e. each value in a signature is 2 bytes long and the index variable used for synchronization is 1 byte long). The public key sent to the receivers is $t * 2 = 512$ bytes long, and sent only once. Therefore, the overhead is $\frac{512 + (12)1290}{30(1290)} * 100\% = 41.32\%$. It is important to notice that the overhead of the HORSE protocol will increase with periodic sending of initial public keys to improve bootstrapping probability. The overhead of the BiBa protocol with the same parameters in the best case is equal to $\frac{512 + (8)30\nu d}{30\nu d} * 100\% = 30.08\%$;(with $d = 20$, and $\nu = \lfloor \frac{t\gamma}{k} \rfloor = 25$; typically $\gamma = 0.10$). The best case for BiBa that achieves the above overhead is when the maximum number of messages is signed in each time interval. That is, for the above parameters in each of the $d = 20$ time intervals, $\nu = 25$ messages are signed. The number of signed messages in each interval is expected to be less than the maximum hence the actual overhead will be larger. The increase in overhead is a result of is inefficient use of keys in BiBa. If the BiBa instances are bootstrapped multiple times to improve successful reception of messages, the overhead is even higher.

The computations that the receivers have to perform when ECDSA is employed for authentication are extensive. In general, public key cryptography are computationally extensive because of the arithmetic involved in signature generation and verifications. Public key cryptosystems perform complex operations on relatively large numbers (160 bits for ECDSA). The PCTs are expected to employ low power microcontrollers with limited computational power. Implementations of ECDSA on an 8-bit processor take 2.78 seconds to verify a signature [24]. On a slightly more powerful 16-bit M16C microcontroller ECDSA was shown to require 630 msec to verify a signature [26]. Comparisons between DSA and HORSE on a PPC 867 MHz platform show that a HORSE implementation using MD5 with $t = 256$, $k = 16$, $d = 2^{10}$ has a key generation time of 0.63 seconds and can verify 2688 signatures in one second [38]. An equivalent DSA has a key generation time of 2.66 seconds and can verify 108 signatures in one second [38] (note that ECDSA is based on DSA). The time required for signature verification in ECDSA is very similar to DSA as shown in [35]. [35] compares software implementations of ECDSA and DSA on a Pentium Pro 200 MHz-based PC to give a perspective on relative performance of ECDSA and DSA. The results show that ECDSA requires 26 ms to verify a signature while DSA require 28.3 ms to verify a signature. HORSE gives a faster signature verification time than DSA (on which ECDSA is based) and a designer faces a tradeoff between the two schemes. The fast verification time of HORSE is ideal for PCT's which are expected to employ microprocessors with modest computational power. The use of HORSE allows for relatively cheap devices at the cost of longer periods to bootstrap the devices. ECDSA does not require bootstrap but requires the devices to perform complex operations and hence take longer times to verify signatures. The time taken to verify a signature could be exploited by an attacker, allowing a denial-of-service attack. If the receivers take a long time to verify a signature, an attacker who floods the network with messages will take up resources at the receivers while they verify the signatures. This could result in missing legitimate messages while the receivers are verifying the bogus messages. Moreover battery powered devices would be unable to go into power-saving modes because they would be kept busy verifying bogus messages, ultimately leading to reduced battery life and increased down times of the receivers.

Security scheme	Network Overhead	95% service availability (KM)	security level (probability of guessing a valid signature)	computational effort (at receiver)
ECDSA (SHA-1)	133.33%	120	2^{-80}	High
HORSE ($t = 512$ and $k = 4$)	41.32%	130	2^{-35}	Low
BiBa ($t = 512$ and $k = 4$)	30.08%	120	2^{-35}	Low

Table 6.2: Comparisons of BiBa, HORSE,and ECDSA

Comparisons of the security schemes are presented in Table 6.2. The table shows the bandwidth overhead introduced by the different security schemes. ECDSA has the most overhead compared to the other schemes. The overhead of HORSE and BiBa as presented in the table is a lower bound and in practice will be higher depending on the number of times public keys are sent to the receivers. The use of signatures increase message size which in turn results in increased message loss-rates. Distances from the transmitter at which 95% of the messages are correctly received with the network parameters set as in Table 6.1 are also presented from simulations. Up to a distance of 90 km all the security schemes achieve high availability of services with a 99.999% probability of receiving messages. Beyond that range, messages may not be received correctly. The loss of messages is due to signatures and failure to bootstrap receivers for BiBa and HORSE protocols. The security level provided by the schemes in terms of successfully forging a signature by guessing is presented in Table 6.2. ECDSA gives a high level of security with a cost of added computational complexity at the receivers. BiBa and HORSE offer reduced security and lower computational cost at the receivers. Reversing the one way hash function used in ECDSA, BiBa and HORSE should be computationally infeasible, hence strong hashing algorithms should be employed. An attacker who cannot obtain private key material from the transmitter and tries to forge a signature is limited to guessing. The probability that an attacker can successfully guess an ECDSA signature is 2^{-80} while for HORSE and BiBa it is 2^{-35} [29].

6.6 Conclusions

We have shown that security (in the form of message authentication) can be offered over the RBDS system. The security provided by the three protocols provides source authentication at a cost of bandwidth efficiency. The construct allows for seamless operation of devices after initialization. The use of a long-term BiBa and HORSE instance allows devices that were rebooted to resynchronize to current short-term instances. The large size of the public keys of the BiBa and HORSE protocol affects the performance, which is dependent on the successful bootstrapping of devices. Although the transmission of the public key for both BiBa and HORSE may take a long time, careful scheduling of such actions for periods with minimal traffic can improve performance. Comparisons shows that the performance of ECDSA is comparable to HORSE. HORSE allows to adjust key sizes and signature sizes to improve performance and reduce overhead. ECDSA on the other hand gives a fixed signature size and introduces significantly higher message overhead compared to BiBa and HORSE.

The protocol employing BiBa described here places constraints on the application data rate. The BiBa protocol assumes an upper bound of 20 messages per day by the application. A single BiBa instance can be used to sign a finite number of messages in a single time interval before the security of the protocol falls below targeted levels. In the event that an application exceeds the number allowed by the protocol within a time interval, a decision needs to be made to either buffer such messages until the next period, send the messages unsigned, or sign the message, albeit with reduced security. Buffering some messages may not be ideal for real-time messages, while some critical messages require authentication at the receivers. A more relaxed design is recommended to avoid placing tight constraints on the application. A tradeoff between the public key sizes and the bound on the application message generation rate exist. Smaller public keys mean fewer messages can be signed in a single period. Allowing for many messages to be sent in a single period requires that either multiple short-term BiBa instances run in parallel or one short-term BiBa instance uses a large number of SEAL chains. The results of both choices effectively increases the size of the commitment keys and ultimately results in increased bootstrapping times and a lowered probability of successful bootstrapping of receivers.

The problem of key distribution still persists with the methods discussed. In the event that a long-term secret key is compromised, there is no feasible way at present to renew it for all the protocols presented. The difficulty is inherent due to the asymmetric nature of the channel. Traditional key agreement and distribution protocols cannot be employed on a one-way communication channel like RBDS. Further studies should be conducted to propose manageable ways of system recovery in the event of a long-term chain compromise.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Our research shows how service provider can support demand response applications over the RBDS network. We present effective device(s) addressing and secure message delivery with the associated tradeoffs. The simulation results presented in our study presents the PCT system as a test case but can easily be extended to a general application that requires broadcasting messages to many receivers. Although the model was calibrated for the Ottawa region, the relative performance and insights obtained apply for other places even though the numbers may be different.

Our study to characterize the RBDS network confirms that RBDS is a viable option to deliver messages to home devices. Our simulation results are consistent with an independent feasibility study on RBDS and PCT communications [8]. The study shows that message reception probability is inversely proportional to message size. Message retransmissions can be used to improve the probability of reception. Simulations show that messages can be delivered to receivers 140km away with 95% reception probability. Thus it is well suited for delivering messages in both urban and rural areas.

Addressing of pervasive devices has been examined to identify concepts that can be adopted to deliver demand response message to home devices for smart grid applications. Our work proposes a compact and efficient addressing mechanism that allows flexible addressing of devices based on locality and logical association of devices. The proposed geographical area address allocation makes for efficient use of limited bandwidth by allowing small and large groups of devices to be targeted easily. The addressing scheme allows each network operator a great degree of freedom to address devices efficiently and it is not strictly limited to a particular network technology.

The security provided by the three protocols in this study provides strong source authentication at a cost of bandwidth efficiency. All three proposals allow for seamless operation of devices after initialization with some initial key material. The use of a long-term BiBa and HORSE instance allows devices that were rebooted to resynchronize to current short-term instances. The large size of the public keys of the BiBa and HORSE protocol affects the performance, which is dependent on successful bootstrapping (i.e., reception of the appropriate short-term key chain instance) of devices. Although the transmission of public key for both BiBa and HORSE may take long times, careful scheduling of such actions for periods with minimal traffic can improve performance. Simulations show that the performance of ECDSA is comparable to HORSE. HORSE allows to adjust key sizes and signature sizes to improve performance and reduce overhead. ECDSA on the other hand gives a fixed signature size and introduces significantly higher overhead compared to BiBa and HORSE.

7.2 Future Work

Research on feasible two-way communication for PCDs has to be undertaken to support smart grid applications that require two-way communications. The security constructs that have been discussed in this study do not have an effective way of distributing a new public key in the event of a compromised private key. There is need to provide for efficient key distribution and management for the security scheme adopted. Testbed trials of the security constructs on the RBDS network need to be carried out before adoption for use.

Bibliography

[1] Technical Review of Residential Programmable Communicating Thermostat Implementation for Title 24-2008. PIER Final Project Report. CEC-500-2007-XXX, Prepared By: University of California, Berkeley Prepared For: California Energy Commission, Public Interest Energy Research Program (CEC-PIER); available at :http://uc-ciee.org/dretd/documents/PCT [cited at p. 41, 44, 104]

[2] United States RBDS Standard. National Radio Systems Committee. Online resource available at : ftp://ftp.rds.org.uk/pub/acrobat/rbds1998.pdf, April. [cited at p. ix, x, 22, 23, 49, 50]

[3] IEEE-SA Technical Report on Utility Communications Architecture (UCA) Version 2.0. The Institute of Electrical and Electronics Engineers, Inc., November 1999. [cited at p. x, 53, 54, 55, 57, 58]

[4] Services and Protocols for Advanced Networks (SPAN): The Structure of the TETRA Numbering Resource, Interworking and High Level Policy for Administration. European Telecommunications Standards Institute, ETSI, May 2003. [cited at p. 57]

[5] The International Public Telecommunication Numbering Plan. International Telecommunication Union, Series E: Overall Network Operation, Telephone Service, Service Operation and Human Factors, February 2005. [cited at p. 57]

[6] Assessment of Demand Response and Advanced Metering Staff Report. Federal Energy Regulatory Commission, available at: http://www.ferc.gov/legal/staff-reports/demand-response.pdf, August 2006. [cited at p. 10]

[7] Security Characteristics of the Title-24 PCT System, 12 April 2007. [cited at p. 3, 69, 70, 71, 73, 74, 75, 76, 77, 78]

[8] Demand Response Research Center RDS-PCT Technology Evaluation (DRAFT). Heschong Mahone Group, Inc., http://h-m-g.com/projects.htm, April 2008. [cited at p. 5, 40, 122]

[9] Integrating New and Emerging Technologies into the California Smart-Grid Infrastructure :A Report on a Smart Grid for California. PIER Final Project Report. CEC-500-2008-048, Prepared By: Electric Power Research Institute (EPRI) Prepared For: California Energy Commission, Public Interest Energy Research Program, September 2008. [cited at p. ix, 9]

[10] Perrig Adrian, Canetti Ran, Tygar J. D., and Song Dawn. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5:2002, 2002. [cited at p. 82]

[11] Liu An and Ning Peng. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 245–256, April 2008. [cited at p. 100, 102]

[12] Juels Ari and Stephen Weis. *Advances in Cryptology - CRYPTO 2005*, volume 3621/2005. Springer Berlin / Heidelberg, 2005. [cited at p. 78]

[13] R.E. Brown. Impact of Smart Grid on Distribution System Design. *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pages 1–4, July 2008. [cited at p. 45]

[14] Alain Chardon, Oskar Almen, Phillip Lewis, Jessica Stromback, and Bertrand Chateu. Demand Response: A Decisive Breakthrough for Europe. Capgemini Consulting; Energy, Utilities and Chemicals in collaboration with Vaata ett and Enerdata. [cited at p. 11]

[15] Knox D.A. and Kunz T. RF Fingerprints for Secure Authentication in Single-Hop WSN. *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing*, pages 567–573, 12-14 October 2008. [cited at p. 80]

[16] Johnson Don, Menezes Alfred, and Vanstone Scott. The Elliptic Curve Digital Signature Algorithm ECDSA. Springer-Verlag, July 21 2001. [cited at p. 100, 101, 102, 103, 113]

[17] Kevin Fall and Kannan Varadran. The NS Manual. The VINT project, available at: http://www.isi.edu/nsnam/ns/, July 2008. [cited at p. 30, 32, 33]

[18] Martin FeldHofer. *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156/2004. Springer Berlin / Heidelberg, 2004. [cited at p. 78]

[19] Taylor Gabriel D., Hungerford David, Rhyne Ivin, and Tutt Tim. Proposed Load Management Standards. Carlifornia Energy Commission, November 2008. [cited at p. ix, 1, 9, 10, 11]

[20] Paul Gibson. Keeping SCADA Open and Secure, June 2008. [cited at p. 78, 79]

[21] Marc Greis. ns Tutorial. available at: http://www.isi.edu/nsnam/ns/tutorial/, cited: June 2008. [cited at p. 31]

[22] Eric Gunther. A Strawman Reference Design for Demand Response Information Exchange. EnerNex Corporation, October 2004. [cited at p. ix, 2, 4, 15, 91, 104]

[23] Eric W. Gunther. Reference Design for Programmable Communicating Thermostats Compliant with Tite 24-2008. CEC PIER PCT Reference Design; available at: http://drrc.lbl.gov/pct/index.html, 26 March 2007. [cited at p. ix, 14, 16, 17, 18]

[24] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. *Cryptographic Hardware and Embedded Systems - CHES 2004*, pages 119–132, 2004. [cited at p. 116]

[25] Patti Harper-Slaboszewicz. FM Radio Offers Intriguing Demand Response Solution. available at: http://www.e-radioinc.com/umc.htm, cited: May 20, 2009, November 2005. [cited at p. ix, 20, 21]

[26] Toshio Hasegawa, Junko Nakajima, and Mitsuru Matsui. A Practical Implementation of Elliptic Curve Cryptosystems over GF(p) on a 16-bit Microcomputer. In *PKC '98: Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography*, pages 182–194, London, UK, 1998. Springer-Verlag. [cited at p. 116]

[27] Ari Juels. RFID Security and Privacy: A Research Survey. *Selected Areas in Communications, IEEE Journal on*, 24(2):381–394, February 2006. [cited at p. 78]

[28] Daucher Micheal, Gartner Eduard, Cortler Micheal, Keller Werner, and Kuhr Hans. RDS-Radio Data System; A Challenge and a Solution. *Fujitsu Ten Technical Journal*, 30 November 2008. [cited at p. 22]

[29] Adrian Perrig. The BiBa One-time Signature and Broadcast Authentication Protocol. *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 28–37, 5-8 November 2001. [cited at p. 83, 85, 86, 92, 119]

[30] Venkat Pothamsetty and Saadat Malik. Smart Grid: Leveraging Intelligent Communications to Transform the Power Infrastructure. Cisco White Paper, available at: www.cisco.com/web/about/citizenship/environment/docs/sGrid_wp_c11-532328.pdf, cited: May 20, 2009, February 2009. [cited at p. 8]

[31] J. Punoose Ratish, Vikitin Pavel, and Stancil Daniel. Effecient Simulation of Ricean Fading in a Packet Simulator. *Vehicular Technology Conference*, 2000. [cited at p. 33, 41, 136]

[32] Borenstein Serverin, Jaske Micheal, and Rosenfeld Arthur. Dynamic Pricing, Advanced Metering, and Demand Response in Electricity Markets. University of Carlifornia Energy Institute, available at: http://repositories.cdlib.org/ucei/csem/CSEMWP-105, 2002. [cited at p. 11]

[33] Engberg Stephan J., Harning Morten Borup, and Jensen Christian Damsgaard. Zero-knowledge Device Authentication: Privacy & Security Enhanced RFID Preserving Business Value and Consumer Convenience. In *PST*, pages 89–101, 2004. [cited at p. 81]

[34] C. Bono Stephen, Green Matthew, Stubblefield Adam, Juels Ari, Rubin Aviel D., and Szydlo Micheal. Security Analysis of a Cryptographically Enabled RFID Device. *14th USENIX Security Symposium*, pages 1–16, August 2005. [cited at p. 70]

[35] Wollinger Thomas, Guajardo Jorge, and Paar Christof. Cryptography in Embedded Systems: An Overview. *Proceedings of the Embedded World Exhibition and Conference*, pages 735–744, February 18-20 2003. [cited at p. 117]

[36] Mander Todd, Cheung Helen, Hamlyn Alexander, and Cheung Richard. Communication Security Architecture for Smart Distribution System Operations. *Electrical Power Conference, 2007. EPC 2007. IEEE Canada*, pages 411–416, 25-26 October 2007. [cited at p. 78]

[37] J. Urbaniak and Paul Harvey. Generation 1 Title 24 Message Format for SMUD Pilot Project, Revision C, February 2008. [cited at p. 24, 25, 26]

[38] Neumann W.D. HORSE: An Extension of an r-Time Signature Scheme With Fast Signing and Verification. *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, 1:129–134 Vol.1, April 2004. [cited at p. 93, 96, 117]

[39] Scott Wright. RBDS version RDS : What are the Differences and How Can Receiver Cope With Both Systems. National Radio Systems Committee, January 1998. [cited at p. 21]

Appendices

Appendix A

Simulation Results

The following table presents the simulation results for the initial study. The results presents message reception probability against distance, message size and number of retransmission. 95% confidence interval and standard deviation of the average values are also given from carrying 10 replications. The results were used to create graphs presented in Chapter 4 of the main document.

interval	real-time mode	normal mode NTX_new = 200s	normal mode NTX_new = 400s
0.1000	0.0000	0.0994	0.0500
0.5000	0.0000	0.4993	0.2492
0.6600	0.0000	0.6646	0.3333
1.0000	0.0000	0.9986	0.4970
10.0000	100.0000	9.9844	5.0078
50.0000	100.0000	100.0000	24.7244
100.0000	100.0000	99.8411	50.1587
200.0000	100.0000	99.6774	100.0000
400.0000	100.0000	100.0000	100.0000

Table A.1: Initial Study Datarate

	80 KM	90KM	100KM	110KM	120KM	130KM	140KM	150KM	160KM	170KM	180KM	190KM	200KM
4B msgs													
1 trans													
Average	89.9900	86.8834	84.4072	81.5408	78.6043	75.9580	72.9415	70.4752	67.9940	65.6028	62.7414	61.1656	59.1546
Std Dev	0.5153	0.5160	0.8802	0.9371	0.7059	0.3940	0.6482	0.9771	1.1141	1.0382	0.8168	1.1383	1.1216
95% C.I.	0.3194	0.3198	0.5455	0.5808	0.4375	0.2442	0.4017	0.6056	0.6905	0.6435	0.5063	0.7055	0.6952
30B msgs													
5 retrans													
Average	100.0000	99.9700	99.9400	99.8799	99.7749	99.5298	99.1846	98.7194	98.0090	96.6383	95.5628	94.0470	91.9910
Std Dev	0.0000	0.0422	0.0568	0.0633	0.1112	0.1136	0.2287	0.3555	0.2235	0.4144	0.5655	0.5820	0.5436
95% C.I.	0.0000	0.0261	0.0352	0.0392	0.0689	0.0704	0.1417	0.2203	0.1385	0.2568	0.3505	0.3607	0.3369
8B msgs													
5 retrans													
Average	100.0000	99.9850	99.9950	99.9450	99.9050	99.8599	99.7949	99.6448	99.2946	98.9795	98.6493	97.9690	97.3337
Std Dev	0.0000	0.0242	0.0158	0.0284	0.0865	0.0967	0.0644	0.1908	0.2422	0.2337	0.2096	0.4017	0.4399
95% C.I.	0.0000	0.0150	0.0098	0.0176	0.0536	0.0599	0.0399	0.1182	0.1501	0.1448	0.1299	0.2490	0.2727
30B msgs													
10 retrans													
Average	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	99.9950	99.9900	99.9650	99.9398	99.9250
Std Dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0158	0.0211	0.0412	0.0616	0.0486
95% C.I.	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0098	0.0208	0.0343	0.0615	0.0409
30B msgs													
2 retrans													
Average	92.6363	88.1291	83.2116	77.5338	70.8754	65.2426	58.0590	50.7854	44.5923	39.9500	34.4772	29.3430	25.9880
Std Dev	0.4917	0.5877	0.9324	0.8242	1.1665	1.0153	0.9480	0.8086	1.1418	0.9591	1.2751	1.1158	0.6892
95% C.I.	0.3048	0.3642	0.5779	0.5108	0.7230	0.6293	0.5876	0.5012	0.7077	0.5945	0.7903	0.6915	0.4272

Table A.2: Initial Performance Results

The following table presents the security performance evaluation simulation results. The data gives average values of message reception probability over distance. 95% confidence intervals of the average packet reception obtained from 100 replications are also presented along with the standard deviation. The data presented here was used to create graphs presented in Chapter 6 in the main document.

	80	90	100	110	120	130	140	150	160	170	180	190	200
No Security													
Average	99.9745	99.9509	99.9469	99.8919	99.7698	99.5786	99.1441	98.6907	97.8742	96.8999	95.5145	93.7708	91.5986
Std Dev	0.0423	0.0641	0.0847	0.1309	0.1482	0.1840	0.2515	0.2993	0.3456	0.5013	0.7195	0.9359	0.8976
95% CI	0.0133	0.0151	0.0166	0.0257	0.0290	0.0361	0.0493	0.0587	0.7853	0.0983	0.1410	0.1834	0.1759
ECDSA													
Average	99.9950	99.9800	99.8899	99.8798	99.0531	98.4014	97.3574	96.7027	95.0400	91.0911	89.2723	87.8308	81.5656
Std Dev	0.0844	0.0986	0.1060	0.2345	0.3637	0.3486	0.5304	0.5664	0.6902	0.8062	1.0424	1.1748	1.3491
95% CI	0.0134	0.0176	0.0208	0.5789	0.0713	0.0683	0.1040	0.1110	0.1353	0.1580	0.2043	0.2303	0.2644
BiBa													
(k=4; t=256)													
Average	99.9940	99.9705	99.9125	98.8209	99.6298	98.3002	92.9245	93.5698	88.1536	88.3687	79.7859	70.4777	65.5413
Std Dev	0.0239	0.0390	0.0733	9.9824	0.1357	9.9308	23.5963	19.3793	27.9141	24.5282	31.4984	37.9327	37.6324
95% C.I.	0.0047	0.0076	0.0144	1.9565	0.0266	1.9464	4.6248	3.7983	5.4711	4.8074	6.1736	7.4347	7.3758
BiBa													
(k=4; t=128)													
Average	99.9915	99.9620	99.9320	96.8529	97.6558	97.3697	89.7054	92.7164	91.0190	86.1816	85.6583	73.8074	72.6923
Std Dev	0.0225	0.0356	0.0490	17.1188	14.0216	13.9809	28.4078	21.5408	23.2082	28.8998	25.7744	35.2006	30.7762
95% C.I.	0.0044	0.0070	0.0096	3.3552	2.7482	2.7402	5.5678	4.2219	4.5487	5.6643	5.0517	6.8992	6.0320
BiBa													
(k=8; t=256)													
Average	99.9800	99.9560	99.9040	96.7994	99.5643	99.2181	92.4202	87.8049	91.2886	86.2971	77.1586	67.1211	62.5743
Std Dev	0.0348	0.0473	0.0673	17.1095	0.1186	0.1825	23.5311	29.4681	21.1832	25.9482	34.0699	37.2632	36.0627
95% C.I.	0.0068	0.0093	0.0132	3.3534	0.0232	0.0358	4.6120	5.7757	4.1518	5.0858	6.6776	7.3034	7.0682
BiBa													
(k=4; t=512)													
Average	99.9885	99.9730	99.9400	99.8219	99.6653	99.2496	98.7794	94.8999	88.4737	85.9280	75.2076	73.6138	64.5523
Std Dev	0.0265	0.0372	0.0498	0.0941	0.1243	1.2722	1.2743	16.9042	26.4222	27.4658	34.1608	34.9872	37.9472
95% C.I.	0.0052	0.0073	0.0098	0.0185	0.0244	0.2494	0.2498	3.3132	5.1787	5.3832	6.6954	6.8574	7.4375
HORSE													
(k=4; t=256)													
Average	99.9850	99.9680	99.9035	99.8019	99.5703	99.2546	98.7499	97.6653	93.0410	90.4387	86.1746	86.3497	77.8659
Std Dev	0.0297	0.0405	0.0874	0.1117	0.1665	0.1833	0.1930	3.0561	17.2461	14.7907	14.4600	12.6715	20.6183
95% C.I.	0.0058	0.0079	0.0171	0.0219	0.0326	0.0359	0.0378	0.5990	3.3802	2.8989	2.8341	2.4836	4.0411
HORSE													
(k=8; t=256)													
Average	99.9910	99.9570	99.9110	99.4437	98.8749	98.5013	96.9060	95.1821	92.5808	87.3592	81.5993	76.0880	71.3392
Std Dev	0.0193	0.0508	0.0724	2.1629	2.9977	3.1075	4.6458	5.4126	6.9508	11.1753	11.9849	13.3948	13.7803
95% C.I.	0.0038	0.0100	0.0142	0.4239	0.5875	0.6091	0.9106	1.0609	1.3623	2.1903	2.3490	2.6253	2.7009
HORSE													
(k=4; t=512)													
Average	99.9930	99.9700	99.9120	99.8064	97.0930	97.7459	92.2036	91.0620	84.1366	80.0665	79.0205	72.8761	66.3972
Std Dev	0.0188	0.0326	0.0715	0.0878	11.1602	8.7375	19.6262	21.1056	27.3727	28.9028	29.6738	29.2293	25.9248
95% C.I.	0.0037	0.0064	0.0140	0.0172	2.1874	1.7125	3.8467	4.1366	5.3649	5.6649	5.8160	5.7288	5.0812

Table A.3: Security Simulation Results

Appendix B

RBDS Group Types

The group types as represented by the Group Type code in the RBDS group as defined in the RBDS standard.

Group type	Group type code	Version	Description
0A	0 0 0 0	0	Basic tuning and switching information only
0B	0 0 0 0	1	Basic tuning and switching information only
1A	0 0 0 1	0	Program Item Number and slow labeling codes only
1B	0 0 0 1	1	Program Item Number
2A	0 0 1 0	0	RadioText only
2B	0 0 1 0	1	RadioText only
3A	0 0 1 1	0	Applications Identification for ODA only
3B	0 0 1 1	1	Open Data Applications
4A	0 1 0 0	0	Clock-time and date only
4B	0 1 0 0	1	Open Data Applications
5A	0 1 0 1	0	Transparent Data Channels (32 channels) or ODA
5B	0 1 0 1	1	Transparent Data Channels (32 channels) or ODA
6A	0 1 1 0	0	In House applications or ODA
6B	0 1 1 0	1	In House applications or ODA
7A	0 1 1 1	0	Radio Paging or ODA
7B	0 1 1 1	1	Open Data Applications
8A	1 0 0 0	0	Traffic Message Channel or ODA
8B	1 0 0 0	1	Open Data Applications
9A	1 0 0 1	0	Emergency Warning System or ODA
9B	1 0 0 1	1	Open Data Applications
10A	1 0 1 0	0	Program Type Name
10B	1 0 1 0	1	Open Data Applications
11A	1 0 1 1	0	Open Data Applications
11B	1 0 1 1	1	Open Data Applications
12A	1 1 0 0	0	Open Data Applications
12B	1 1 0 0	1	Open Data Applications
13A	1 1 0 1	0	Enhanced Radio Paging or ODA
13B	1 1 0 1	1	Open Data Applications
14A	1 1 1 0	0	Enhanced Other Networks information only
14B	1 1 1 0	1	Enhanced Other Networks information only
15A	1 1 1 1	0	Defined in RBDS only
15B	1 1 1 1	1	Fast switching information only

Table B.1: RBDS Group Types

Appendix C

Modifications to NS-2

This section presents the changes made to the default NS-2 files and the new files created for modeling the RBDS network. A brief discussion of the changes made to the files are presented here. This section outlines the files necessary to reproduce the work presented in this thesis. The source code is available on request from the author. The following files have to be imported into the NS distribution and added to the Makefile for compilation. It is important to note that the distribution used to create, test and use the files is NS-2.30. A complementary readme file will be provided along with the distribution and some supporting sample scripts. The following are important files that are required if one wants to export relevant source code only and use in a different NS distribution.

- **prop_ricean_shadowing.cc and prop_ricean_shadowing.h**

  These are new files and should be placed in the NS-2 Mobile directory. The model described in the files is a modification of the default Ricean/Rayleigh fast fading model which uses the TwoRayGround as the large scale propagation model. Changes from the original Ricean/Rayleigh model include replacing the TwoRayGround propagation model with the Shadowing model. The model logs the physical layer output to a separate trace file specified in the OTcl script using the instruction:

```
set val(proplog) proplog.tr ;#Set Log for RF propagation info
```

The logged data format is the same as the default Ricean/Rayleigh model (see [31]). Refer to the next appendix for usage and setting of parameters of the model.

- **mac-rds.cc and mac-rds.h**

  These are new files that define the RBDS media access and should be placed in the Mac directory. The RBDS message fragmentation and construction from fagments is done here. See next appendix on usage of the model.

- **biba.cc and biba.h**

  New files that define the dynamics of the BiBa signature protocol and message types. The parameters of the protocol such as chain lengths and chain sizes can be configured through the biba.h file. The files have to be added in the mac directory of NS-2.

- **horse.cc and horse.h**

  New files that define the dynamics of the HORSE signature protocol and message types. The parameters of the protocol such as chain lengths and chain sizes can be configured through the horse.h file. The files have to be added in the mac directory of NS-2.

- **ll.cc and ll.h**

The changes made to the ll.h and ll.cc are to declare BiBa and HORSE classes as members of the LL class. This allows the LL class to use the security features when sending (encrypting) and receiving (decrypting) messages. This file has to be updated to allow the security features. Refer to the distribution source to update it.

- **dumbagent.cc**

  Default file located in the NS-2 Mobile directory requires updating to keep up with the distribution to pass all messages up and down the protocol stack.

- **title-24.cc and title-24.h**

  This is a simple application that generates PCT messages after predefined fixed intervals. More complex (statistical) intervals can be defined in this file as necessary. These files should be added in the Application directory of NS-2

- **packet.h**

  The following changes are necessary to accomodate the new packet types that are generated by the security and RBDS models. The packet.h file located in the common directory was modified to define new packet types. The two security protocols were made to use the same packet structure to simplify things. The following two lines were added to packet.h define the packet structures.

```
 #define HDR_BIBA(p) (hdr_biba::access(p))
#define HDR_MAC_RDS(p) ((hdr_mac_rds *)hdr_mac::access(p))
```

  To assign the packets numbers, PT_BIBA,PT_RDS, and PT_TITLE24 have to be added to the *enum packet_t* part of the packet.h. String names can be allocated following the structure and format in the *p_info()*.

Appendix D

Implementation and Usage Of Models in NS-2

The models used in the simulations were implemented in C++. Usage of the models in NS-2 through C++ and OTcl is given below.

D.1 Physical Layer

The physical layer propagation model was modeled as described in Section 4.1.2. An example of the OTcl instructions to use the Ricean/Rayleigh propagation model described in Section 4.1.2 is given below. It is important to configure the large scale fading channel correctly. The NS-2 Shadowing model requires that no two nodes have a separation distance of 0 otherwise there will be an illegal division by zero exception.

```
    set val(prop) Propagation/RiceanShadowing
## Rayleigh and Ricean with Shadowing as the large scale fading
set val(RiceanK) 0.0
## Ricean K factor
set val(RiceanMaxVel) 120
## Ricean Propagation MaxVelocity Parameter
Propagation/RiceanShadowing set pathlossExp_ 3.5
## Path loss exponent for large scale fading
Propagation/RiceanShadowing set std_db_ 12
## Shadowing deviation due to multipath propagation
Propagation/RiceanShadowing set dist0_ 45.125
## Close in distance
Propagation/RiceanShadowing set seed_ [expr {int(rand()*64)}]
## seed for the power deviation random var
## to obtain identical simulation fix this to a number between 0 and 63
## (which indexes one of the 64 pre-defined seeds in ns-2 )
```

The model requires a data file of precomputed values used in the calculation of the fast fading channel. The file is provided with the Ricean/Rayleigh distribution and its location must be specified in the script (an easy way is to keep it in the working directory). The data file can be specified using the following instruction (if the data file is in the same directory as the script, otherwise the absolute address would be necessary):

```
    set val(RiceDataFile) rice_table.txt ;# Ricean Propagation Data File
```

D.2 Media Layer (RBDS)

RBDS is modeled using the class MacRDS class which extends the default NS-2 Mac class. Usage of the RBDS model through the OTcl script is given below. To use the RBDS as the media access layer in a OTcl script, the following line is used:

```
    set val(mac) Mac/RDS
```

RBDS has a raw bitrate of 1187.5 bits per second. The *bandwidth_* variable in the RBDS model can be used to set the mac raw datarate. The two modes of operation of RBDS are selected by the use of a *mode_* variable. A value of 0 means normal mode of operation will be used to schedule application messages, while a value of 1 uses the real-time mode of RBDS. The number of retransmissions of each message is represented by the variable *NTX_repeat_*. The variable can be set through the OTcl script to any integer greater than or equal to 1. The minimum time (in seconds) between successive unique messages can be set through the *DTX_new* in the script. The *DTX_new* variable applies to the normal mode of operation and has no use for the real-time mode. In this RBDS implementation, the node with ID 0 (i.e. the node created first in the script) will be the base station. This is the only node that can send any packets, all other nodes are passive receivers and are not allocated sending slots. An example of setting the parameters for the MacRDS class in OTcl is given below:

```
    Mac/RDS set bandwidth_ 1187.5 ; # RBDS bitrate
Mac/RDS set mode_ 1 ;# RBDS modes:  0= Normal Mode ;1 = Real-Time mode
Mac/RDS set NTX_repeat_ 5 ;# Number of times a message is repeated
Mac/RDS set DTX_new_ 300 ;# Min time between unique successive messages
```

D.3 Security

Security was implemented in the link layer. This was done to decouple it from the medium access (RBDS). The link layer is the layer directly above the mac layer hence our security protocol was implemented in the link layer to simplify things. Our Implementation replaces adds to the default NS-2 LL (link layer) class, a member of either HORSE or BiBa (these instances should be mutually exclusive). The BiBa and HORSE classes are extentions of the NS-2 Process class. Both classes are friend classes of the LL class i.e. they have visibility and can use member methods of the LL class. This allows the classes to send and receive background traffic that they generate through the link layer. When either of the BiBa or HORSE instances are used to provide for security, the *recv()* method of LL invokes *encrypt()* or *decrypt()* methods as appropriate before propagating a received packet along the stack. When the LL receives an outgoing packet from the application, it invokes the *encrypt()* method of the security object before sending the packet to the medium access layer (RBDS) below. When a message is recieved from the lower layers (RBDS), the LL invokes the *decrypt()* method of the security object before passing up to the application. The *encrypt()* method generates a signature and appends it to the message and returns a signed message. The *decrypt()* method verifies the signature and removes it from the message.

A graphical representation of the interactions of the classes is given by Figure D.1. The figure shows how a message is propagated from the application to the channel and back up to the application. When employing a security instance (BiBa or HORSE), a message is signed as it is passed down the communication stack. The RBDS layer fragments the message and sends it out in multiple RBDS fragments. The receiver RBDS layer tries to reconstruct the signed message and passes it up if all fragments are received. Otherwise the message is discarded. When the LL recieves the message, it invokes decryption from the security instance. If the signature cannot be verified, the message is discarded. Otherwise the message is passed up to the application.

To configure the security objects through the OTcl interface one of the following instructions is used. This allows one to choose between HORSE, BiBa and no security in the link layer.

```
    set val(ll) LL/HORSE ;# Selects the HORSE protocol
#set val(ll) LL/BiBa ;# Selects the BiBa protocol
```

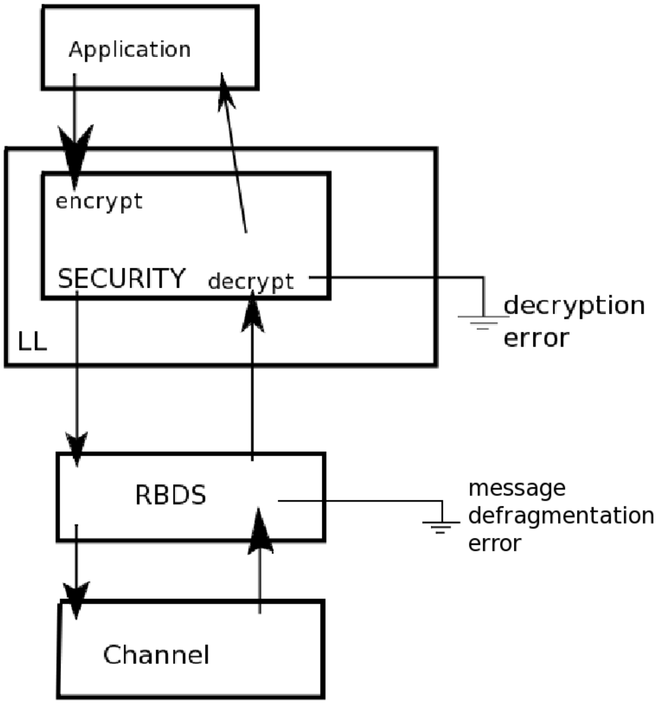


Figure D.1: Interactions of Model Classes

```
#set val(ll) LL ;# No security
```

It is important when using both the BiBa and HORSE security to allow the initial public keys to be transmitted to the receivers before sending application messages. A delay of 2000 seconds simulation time is more than sufficient to allow the receivers to be bootstrapped at the beginning of the simulation. The exact delay time required will depend on the size of keys and number of retransmissions, but 2000 seconds was found to be enough for most settings.

The BiBa model is implemented as described in Section 6.4.1. The class uses timers to update the public keys when the time expires. HORSE model is also implemented as described in 6.4.2. HORSE does not use timers and updates public keys when one of the current chains gets exhausted. It is important to notice that if the public keys are not successfuly received, all messages received in the time period will not be correctly decrypted in both BiBa and HORSE. ECDSA was not implemented exclusively as separate security class, instead messages 40 bytes larger than normal messages were used to model the signatures introduced by ECDSA.

D.4 Application Layer

The application class Title-24 extends the NS-2 Application class. It employs a simple timer that expires after *interval_* seconds. When the timer expires, the application produces an application message *size_* bytes and resets the timer. The variables *interval_* and *size_* can be set through the OTcl script as shown below:

```
    #Create a Title-24 application
set tt24 [new Application/Title-24]
$tt24 set interval_ 4320.0
$tt24 set size_ 30
```