DISSERTATION

A HIGH BANDWIDTH RADAR OPERATION OVER THE INTERNET: SIGNAL ANALYSIS, NETWORK PROTOCOLS AND EXPERIMENTAL VALIDATION

Submitted by

Yoong-Goog Cho

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

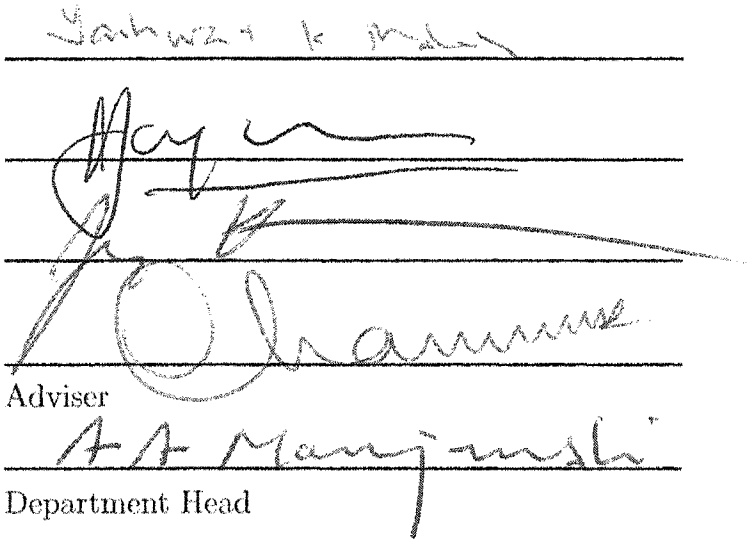
Spring, 2004

COLORADO STATE UNIVERSITY

December 8, 2003

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY YOONG-GOOG CHO ENTITLED A HIGH BANDWIDTH RADAR OPERATION OVER THE INTERNET: SIGNAL ANALYSIS, NETWORK PROTOCOLS AND EXPERIMENTAL VALIDATION BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work



Adviser

Department Head

ABSTRACT OF DISSERTATION

A HIGH BANDWIDTH RADAR OPERATION OVER THE INTERNET: SIGNAL ANALYSIS, NETWORK PROTOCOLS AND EXPERIMENTAL VALIDATION

The Virtual-CHILL (VCHILL) project is an initiative to provide full operation of the CSU-CHILL radar over the Internet. As part of this project, VCHILL aims to transmit Digitized Radar Signal (DRS) in real-time over high-bandwidth data network, estimate radar parameters at remote sites, and distribute them for display and further applications. This concept is referred to as the high-bandwidth VCHILL. The distribution of real-time DRS enables the remote users to process the digitized signals according to their interests.

First, a new digital-IF receiver, which operates in parallel with the existing digital receiver (DRX) of the CSU-CHILL radar, was developed. The parallel receiver operates in conjunction with the DRS network server so that the server arranges the DRS into a specified format in real-time and transmits it to clients at up to a few hundred Mbps. Subsequently, end-system architectures for the high-bandwidth VCHILL were developed. The architecture design includes various functions, such as DRS acquisition, DRS transmission, DRS receive/radar parameter computation, and parameter transmission, as well as generic packet and data structures for the data transmission and sharing. Third, transmission waveform design scheme for congestion control was developed to make the real-time operation tolerable and

provide high quality end products under unpredictable network conditions. The developed scheme relies on the operating principle of the pulsed-Doppler radar and estimation theory of the radar parameters. A combination of this waveform design scheme and a source-based rate control algorithm with Additive Increase and Proportionate Decrease based on the feedback provides the highest quality of service possible. Performance evaluation of the system, while operating the CSU-CHILL radar in real-time, shows the linearity of the end-to-end TCP throughput proportional to the data rate, coincidence with the output of the DRX system, quality improvement of display, as well as apparent congestion control functionality. Finally, to increase computing capacity at client sites, a concept of distributed DRS client was developed. The essential idea is that the computation loads are evenly distributed among the computing nodes that are connected to each other in networked environment.

Another aspect of the VCHILL is networked radar operation. In this scenario, a networked radar system can be readily used for the simultaneous observation of a same target by multiple radars, and the replacement of a large centralized radar with multiple small radars. Operational models, end-system architectures, and network models to make possible the notion of the networked real-time weather radar system are proposed. Scalability of the networked radar system was studied in terms of power consumption for pulse transmission, computation load, and communication load varying the Quality of Service parameters of spatial resolution and data accuracy.

Yoong-Goog Cho
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, Colorado 80523
Spring, 2004

ACKNOWLEDGEMENTS

"But by the grace of God I am what I am, and his grace to me was not without effect. No, I worked harder than all of them - yet not I, but the grace of God that was with me." (1 Gorinthians 15:10)

I express my profound sense of gratitude to Dr. V. Chandrasekar, my advisor, for his encouragement, valuable guidance and for providing me with the opportunity to pursue my aspirations at Colorado State University. My gratitude goes to Dr. Anura P. Jayasumana for his guidance and suggestions. I would also like to express my gratitude to Dr. V. N. Bringi, Dr. John Hubbert and Dr. Yashiwant Malaiya for their serving as committee members.

I do not know how to express my deep appreciation to David Brunkow and Patrick Kennedy, who are senior engineer and facility manager of the CSU-CHILL radar respectively. My dissertation could not have been made possible without their help and concern. I express my thanks to Louie Canaiy and Dr. Gwo-Jong Huang for their assistance. I also would like to extend my thanks to colleagues for their help and assistance: Sang-Hoon Lim, Sutanay Choudhury, Nitin Bharadwaj, Chris R. Rose, Konrad Gojara, Bare Abhijit, and Daniel Vivanco.

I express my deep gratitude to my mother who is praying for me all the time. God who replies to her pray and devotion has protected me and made me keep on going. I also thank to my cousin brother Eui-Chi Jung and his family for taking care of my mother. I would like to express my appreciation to my parents-in-law for their love, encouragement and support. Finally, thanks goes to my family,

my wife Youm-Won Park, son Il-Gyu, and daughter Sun-Joo, for their love and devotion.

DEDICATION

This work is dedicated to my father who passed away four years ago in winter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Brief Overview of VCHILL Project

The Colorado State University (CSU)-CHILL radar system is a national facility operated under a cooperative agreement with the National Science Foundation to support education and research in remote sensing of the atmosphere [1]. The CHILL radar is an S-band dual-polarization Doppler radar with polarization agility and diversity [2].

Virtual CHILL (VCHILL) project is an initiative to enable the real-time operation of radar over the Internet, which is a paradigm shift in comparison with the traditional radar operation [3]. The traditional way for operating the radar is that the researchers visit the radar site or get the data in tapes at the end of data collection process. The VCHILL initiative changes this paradigm as shown in Fig. 1.1. The VCHILL paradigm is being implemented at various levels, such as low-bandwidth VCHILL, high-bandwidth VCHILL, and networked radar operation.

The goal of the low-bandwidth VCHILL is to transfer the integrated radar parameters that are estimated at the radar site to remotely located display nodes with various transport layer protocols over the Internet [4, 5]. Since the transmission requires the bandwidth of a few hundred Kbps, it is referred to as a low-bandwidth VCHILL. Currently many universities and research communities over the United States are able to access and display either the real-time or the archived the CSU-CHILL radar data for their educational or research purpose. They can also control

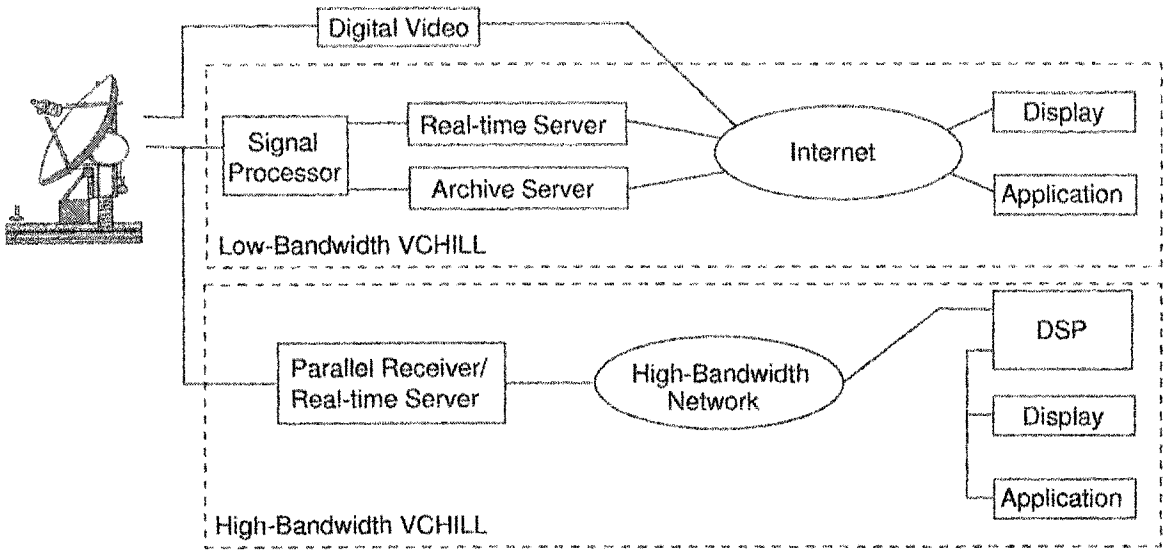


Figure 1.1: A conceptual diagram of the VCHILL.

the radar system remotely through graphic user interface simultaneously observing the radar system with a digital video system [1].

The goal of the high-bandwidth VCHILL is to transmit Digitized Radar Signal (DRS) to remotely located client sites in real-time over high-bandwidth data network, such as Next Generation Internet (NGI) or Gigabit Ethernet, where the radar parameter estimation process is physically and logically separated from the radar. Providing the remote sites with the DRS enables the radar to be operated in a typical multi-user environment wherein each user independently processes the same digitized output signals for various applications as well as with varying algorithms, such as clutter filtering, spectral processing, and oversampling [6]. This approach to radar signal estimation is likely to stimulate development of new and innovative signal-processing algorithms. The high-bandwidth application involves challenges that include strict real-time requirements for manipulating DRS and signal processing, variable latencies introduced by physical distances, router delays and end-system performances, as well as deficiency of the available bandwidth [6].

The goal of the networked radar operation is to develop the networked radar system, which can be used for the simultaneous observation of same targets and the replacement of a large centralized radar with multiple small radars [7]. Operating multiple networked radar systems increases in reliability of the overall radar system and the quality of the end product compared to a single conventional radar, such as high spatial resolution and reduced coverage floor [8, 2]. Moreover, the networked distributed radar system can also provide benefits in radar system design, such as great reductions in cost brought about by low power transmitters, cheaper receivers, as well as smaller antennas, radome and pedestal size [6]. Research towards achieving this comprises the developments of operational models, network architecture and end-system architecture for the operating the real-time networked radar system, including application layer management protocols.

1.2 Objectives of the Research

This dissertation primarily focuses on the high-bandwidth aspect of VCHILL and the networked radar operation. The primary goal of the research is development and experimental validation of the high-bandwidth VCHILL concept. This spans into numerous specific objectives, namely a) evaluation of radar signal handling and networking protocols for transmitting the high-bandwidth digitized radar signal in real-time and b) theoretical studies on the operation of networked radar systems. The accomplishments of this research include the development of the digital-IF radar receiver system, the development of the end-system architectures for the high-bandwidth VCHILL, the development of a transmission waveform design for congestion control for the high-bandwidth VCHILL, as well as the development of the distributed DRS client. In addition, the study on the networked radar system in terms of operating model, network model and end system architecture is included.

1.3 Overview of the Dissertation

The organization of this dissertation is described as follows.

- **Chapter 2: Development of a parallel digital-IF radar receiver for the high-bandwidth VCHILL**

  Chapter 2 describes the parallel digital-IF receiver systems developed exclusively for the high-bandwidth VCHILL. The description includes the functionality of blocks composing the new receiver system, details on the design of the system, and the performance evaluation. The chapter also gives a brief description on the application layer from the perspective of the network server.

- **Chapter 3: Development of end-system architectures for the high-bandwidth VCHILL**

  The end-system architectures for the real-time operation of the high-bandwidth VCHILL are described in Chapter 3. The description includes the design philosophy, operation of the individual process, and the implementation and performance evaluation of the architectures. The output from the two receiver systems are compared. In addition, critical factors affecting the end-to-end TCP performance are explored.

- **Chapter 4: Development of transmission waveform design for congestion control in the high-bandwidth VCHILL**

  Chapter 4 explains data selection schemes and a congestion control algorithm to provide tolerance of the high-bandwidth VCHILL system. The theoretical basis and the simulations to develop the signal handling scheme are explained. Performance results showing the transmission rate and quality of the radar measurements are also presented.

- **Chapter 5: Development of the distributed DRS client**

  The concept and architectures of the distributed DRS client system to achieve the required computing capability are described in Chapter 5. Performance evaluation results to demonstrate the feasibility for the high-bandwidth VCHILL applications are also included.

- **Chapter 6: Study on the networked radar system**

  Chapter 6 explores the operational models, end-system architectures, and network models for enabling the notion of the networked radar system. The chapter also introduces Quality of Service (QoS) parameters for the applications of the networked radar system. Scalability is examined varying the QoS parameters. Finally, technical issues related to the networked radar systems are discussed.

- **Chapter 7: Conclusion**

Chapter 2

DEVELOPMENT OF THE DIGITAL-IF RADAR RECEIVER FOR THE HIGH-BANDWIDTH VCHILL

2.1 Introduction

Essential idea of the high-bandwidth Virtual CHILL (VCHILL) is that the received signals of the CSU-CHILL radar are digitized at the radar site and transferred to remote sites in real-time over a high-bandwidth data network [3, 6]. Once received over the primary network, radar parameters can be estimated at remote sites and distributed to display nodes over other remote data networks [9]. This all-digital networked approach is very useful if the radar is operated in a typical multi-user environment wherein each user independently processes the same digitized output signals for various applications, such as clutter filtering, spectral processing, and oversampling [10]. This networked approach to radar signal processing is likely to stimulate development of new and innovative signal-processing algorithms.

Up until now, the received signals of the CSU-CHILL radar have been routinely processed by the existing digital receiver/processor (Lassen Research Aspen/DRX unit) which produces end products with the conventional radar operation paradigm [2, 11]. Attempts to modify the current mode for the high-bandwidth VCHILL validation, while maintaining the current operation, were not possible. After extensive analysis, an independent receiver that would serve the

high-bandwidth VCHILL system primarily is the best way to implement its concept without extensively compromising numerous innovative aspects of the high-bandwidth VCHILL, such as agility in transmit waveform design and estimation algorithm.

To achieve flexibility in the high-bandwidth VCHILL, a new IF-digital receiver system is designed to operate in parallel with the existing DRX signal processor. Data rate generated by the parallel receiver is determined by a combination of factors relating to quantization level, sampling rate, and number of receive channels. The parallel receiver for dual channels of coherent receiver system sampled at a few MHz and 16-bit quantization level produces an overall data rate of a few hundreds Mbps. The parallel receiver functions as a part of the DRS server concept in networked environment so that it integrates all input data into a specified format in real-time. Because of the features of the pulsed-Doppler polarimetric radar operation and the high data rate, the integration requires extremely careful timing for synchronized data acquisition and high speed data manipulation.

Development of this new digital-IF parallel receiver for the CSU-CHILL radar combined with innovative data manipulation protocol for the transmission of the digitized radar signal (DRS) lays the foundation of the high-bandwidth VCHILL. Another innovative aspect of this development is that the receiver system has been implemented using a generic Analog-to-Digital Conversion/Digital-Downconversion (ADC/DDC) signal processor. The signal processor is interfaced with high-speed PCI slot mounted on a standard PC (Intel Xeon Dual processors 2.8 GHz) running with Linux. Therefore, the performance can be continuously improved by the advent of high speed PC/PCI platforms as well as upgraded signal processor.

2.2 Received signals of polarimetric Doppler radar system

Transmitter of a pulsed-Doppler radar consists of the stabilized local oscillator (STALO), pulse modulator and power amplifier. The STALO generates a contin-

uous wave signal of nearly perfect sinusoid form which is modulated and amplified to produce high power microwave energy. The pulse modulator generates a train of microwave pulses that are spaced at the pulse repetition time (PRT) T_s interval. The transmitted signal can be written as,

$$S_{tr}(t) = U_{tr}(t)\exp(j2\pi f_0 t) \tag{2.1}$$

where f_0 is the carrier frequency, t is time. The rectangular pulse is defined as,

$$U_{tr}(t) = \begin{cases} 1 & : \ -\frac{T_0}{2} \leq t \leq \frac{T_0}{2}, \\ 0 & : \ \texttt{otherwise}, \end{cases} \tag{2.2}$$

where T_0 is the pulse width.

When the STALO is connected to synchronous detectors, the received signal due to a point scatter moving a uniform velocity is proportional to the transmitted waveform, U_{tr}. The received signal is given by,

$$S_r(t) = AU_{tr}(t-\tau)\exp(j2\pi f_0(t-\tau)) \tag{2.3a}$$

$$= A\exp(-j2\pi f_0\tau)U_{tr}(t-\tau)\exp(j2\pi f_0 t) \tag{2.3b}$$

where τ is defined as $2r/c$ with r being range and c being the speed of light, A is the complex amplitude, $\lambda G\sqrt{P_t}S/4\pi r^2$. Here, G is receiver gain, P_t is transmitted signal power, and S is scattering matrix element. Therefore, the echo voltage for a single moving particle is given by,

$$V_r(t) = A\exp(-j2\pi f_0\tau)U_{tr}(t-\tau) \tag{2.4}$$

In general, the scattering amplitude can be time-varying and the functional dependence can be expressed as $A(\tau;t)$. Additionally, precipitation is composed of a large number of hydrometeors extending over a large range with widely different scattering amplitudes and moving with different velocities. Therefore, the received

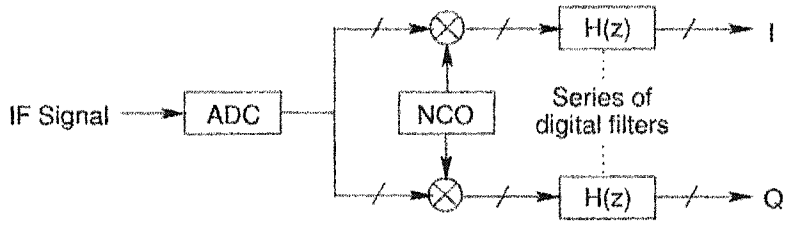


Figure 2.1: A quadrature demodulation scheme.

voltage can be expressed as a discrete sum of the contribution from individual particles in the medium as,

$$V_r(t) = \sum_k A_k(\tau_k; t) \exp(-j2\pi f_0 \tau_k) U_{tr}(t - \tau_k) \tag{2.5}$$

where A_k is the scattering amplitude of the kth particle, and τ_k is $2r_k/c$ [12].

Virtually all radar receivers operate on the superheterodyne principle. The echo signal, after modest amplification, is shifted to an IF by mixing with a local oscillator (LO) frequency. Actual receivers have several stages of RF and IF processing. The received signals of the CSU-CHILL radar are down converted to a 50-MHz IF by mixing the received signal with the STALO frequency after passing through image rejection filters. A second conversion to the 10-MHz IF is performed by mixing with the 40-MHz reference clock coming from the DRX processor [2, 11].

In the digital radar receiver which performs IF digitizing, an IF signal is applied directly to the input of a wide-bandwidth ADC, then digital signal processing (DSP) is used to process the digitized samples. Figure 2.1 shows an example of digital quadrature demodulation scheme [13]. The system consists of digital mixers, numerically controlled oscillators (NCOs), and digital filters. The digitized IF signal is multiplied with cosine and sine wave samples coming from the NCO to extract In-phase (I) and Quadrature (Q) components, respectively. The series of digital filters then perform low-pass filtering and digital decimation to get the digitized signal data at a desired sampling rate.

The CSU-CHILL radar can transmit four types of polarization sequences, namely vertical (V) only, horizontal (H) only, alternating between V and H, and hybrid mode which contains V and H components simultaneously [1]. The dual-polarized radar system also has two coherent receivers, V and H receivers. The receive states are always V and H independent of the transmitting states. The digitized signal at the receiver is referred to as Digitized Radar Signal (DRS). It is well known that the alternating mode and the hybrid mode can provide extensive set of additional parameters compared to the V or H only transmit mode [12].

The CSU-CHILL radar transmits pulses at a PRT of an 800 - 2000 μsec. After transmission of every pulse the receiver digitizes the received signal with a few MHz sampling rate during PRT. At a particular pointing angle, the pulses are transmitted multiple times to obtain reliable data sets. A digitized data set of the return signal from a transmitted radar pulse and the regularly spaced data are referred to as 'a range sample data set' and 'gate', respectively. Therefore, the position in the row of the data set indicates the distance of the object from the radar. Here, we have two types of time scales, one is 'range time' that is associated with the digitizing of the return signals. Another is 'sample time' that is related to the time at which pulses are transmitted. The concepts of the range time and sample time are illustrated in Appendix A.

For the CSU-CHILL radar, four combinations of the received signal can be obtained depending on the radar operating mode as shown in Eq. (2.6). Convention that we will follow to denote the DRS through this dissertation is written as belows.

$$V_{vv}[n] = I_{vv}[n] + jQ_{vv}[n] \tag{2.6a}$$

$$V_{hv}[n] = I_{hv}[n] + jQ_{hv}[n] \tag{2.6b}$$

$$V_{vh}[n] = I_{vh}[n] + jQ_{vh}[n] \tag{2.6c}$$

$$V_{hh}[n] = I_{hh}[n] + jQ_{hh}[n] \tag{2.6d}$$

where V stands for the complex voltage of digitized signal which consists of I and Q components. The first and second subscripts indicate the receiver and the polarization status of the transmitted pulse, respectively. The n within square brackets indicates the n^{th} sample of the received signals with the same range time.

2.3 Design of the digital receiver system

2.3.1 Overview of the digital receiver system

The digital receiver system is designed to operate in parallel with the Aspen/DRX signal digitizer/processor that is the existing system of the CHILL radar. Figure 2.2 provides a schematic diagram showing the various signal sources and the connections associated with the parallel receiver. The receiver system is made up of an ADC/DDC signal processor, a digital Input/Output (I/O) card, and a serial I/O port, as well as a standard PC and a circuit board to convert data format of Azimuth (AZ) and elevation (EL) angles.

The received signals split from the existing channels are fed into the ADC/DDC signal processor, along with the 40-MHz clock and a sampling trigger signal. The processor digitizes the received IF signal, produces I and Q components of the digitized signals, and decimates them by a selected factor. The sampling trigger signal, which is active every time a pulse is transmitted, informs the ADC/DDC signal processor of the instant when it starts to sample. The ADC/DDC signal processor is interfaced with the 64-bit/66-MHz PCI bus of the DRS server.

The signals of Azimuth, elevation angles, and the polarization states of the transmitted pulses go into the digital I/O card with two types of trigger signals coming from the DRX processor. The digital I/O card reads the Azimuth and elevation angles upon a receipt of the integration cycle trigger signal that is active every time the first pulse is transmitted at a particular pointing angle. The Azimuth and elevation angles coming from the position sensor in serial data form are

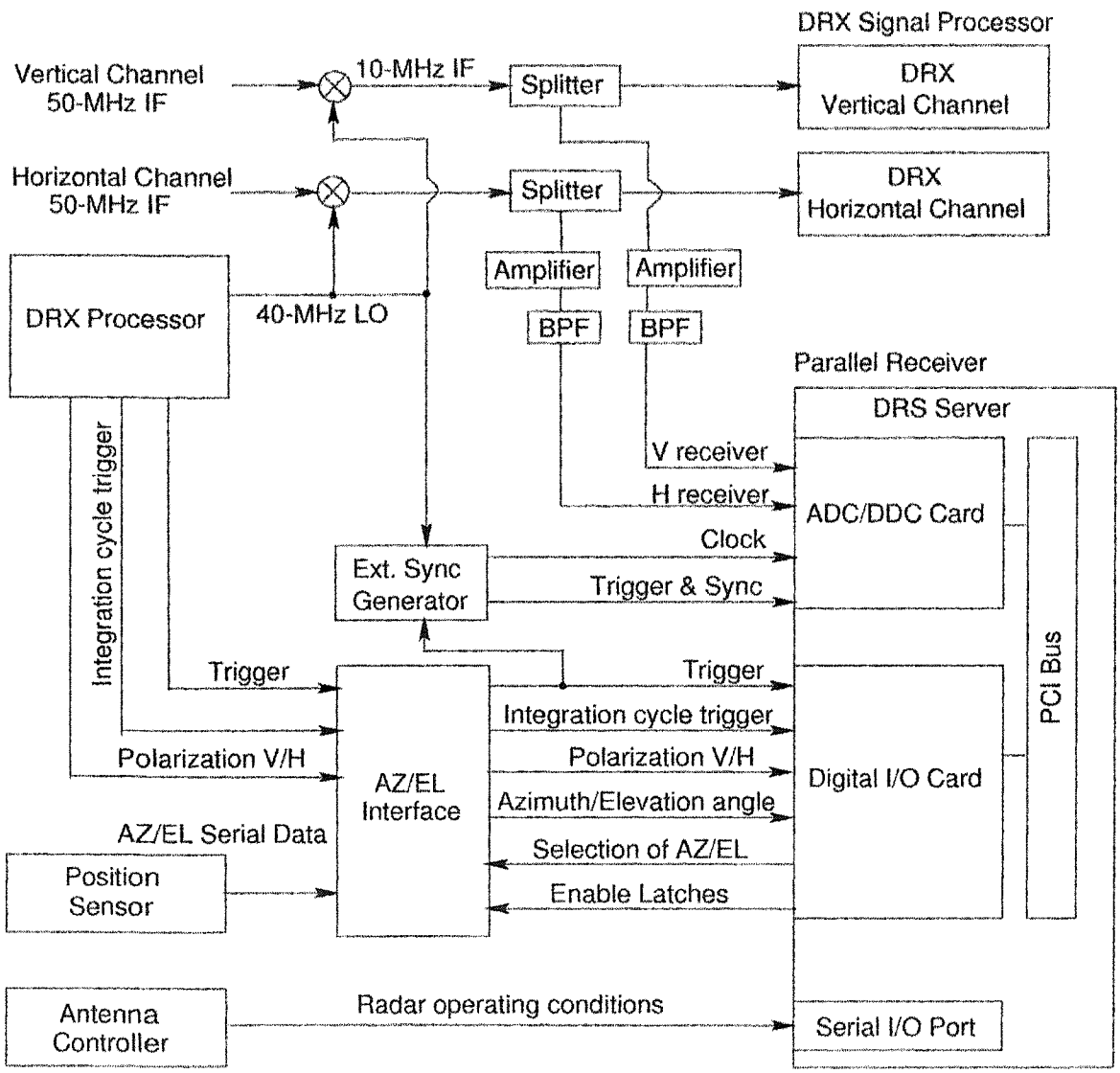


Figure 2.2: Interfaces between the CHILL radar and the DRS server.

converted to parallel data format at the AZ/EL interface circuit. The 16-bit data bus between the AZ/EL interface circuit and the digital I/O card is shared for delivery of both Azimuth and elevation angle data. The digital I/O card is interfaced with the 32-bit/33-MHz PCI bus of the DRS server. Other important radar operating conditions coming from the antenna controller are periodically transferred into the DRS server through serial I/O port. The parallel receiver arranges all these data into a specified format and passes them to a thread that is responsible for transmission to remote sites. Detail information on the data formats will be explained in later sections.

2.3.2 Generic functionality of the ADC/DDC signal processor

The ADC/DDC signal processor from the Interactive Circuits and Systems Ltd. (Model: ICS-554B-MN) has not only 14-bit ADC module, but also DDC capability [14]. It has four input channels and is available with 100 MHz maximum sampling clock per channel. Figure 2.3 shows the block diagram of the signal processor. The full-scale input signal level is approximately 1.2V peak-to-peak (+5.5dBm) into 50 ohms. The fundamental functions of the ICS-554B-MN signal processor and the evaluated results with artificial input signals will be described in this section. Detail information about the signal processor can be found in the instruction manual [14].

When the ADC output mode is selected, the digitized signals bypass the DDC module and are stored in the FIFO as 2's complement 16-bit samples. These are justified to the most significant 14 bits out of each 16-bit word. Figure 2.4 shows the quantized values with increase in the peak-to-peak voltage of the input signal. Figure 2.5 shows the good linearity between output of the ADC and voltage of the input signal.

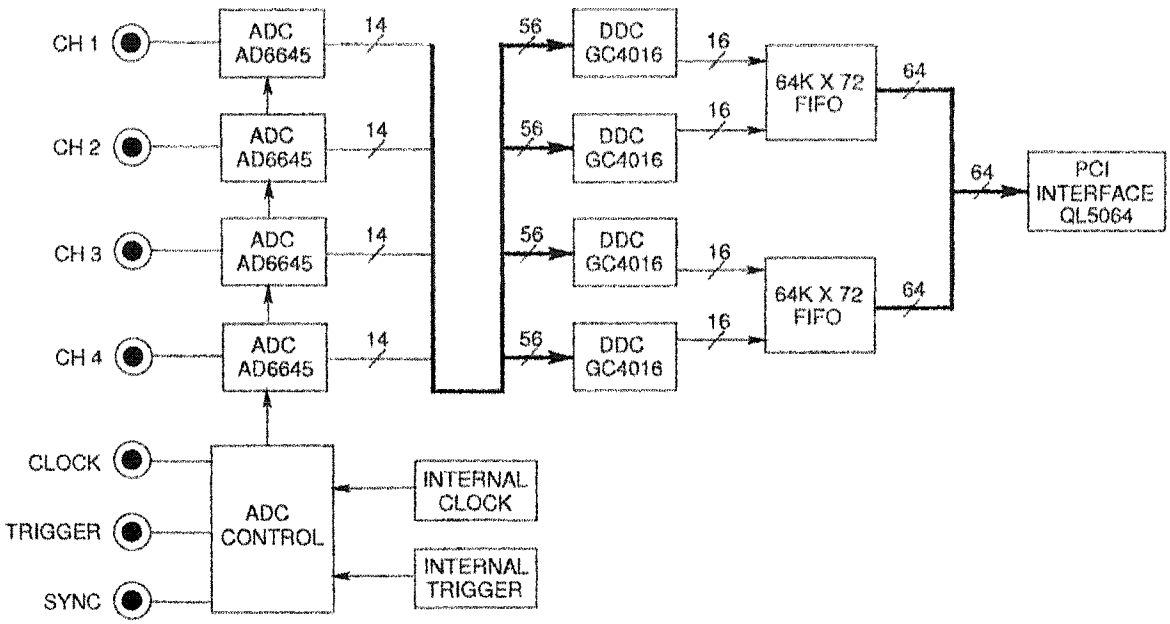


Figure 2.3: Block diagram of the ICS-554B-MN signal processor.

The ICS-554B-MN signal processor includes four Graychip GC4016 DDC chips, each containing four identical digital down-conversion circuits, giving a total of up to 16 individual output channels [15]. Each of the four identical down-converters on the DDC chip accepts a sampling clock up to 100 MHz, down-converts a selected IF frequency, filters the signal, decimates the signal rate by a programmable factor ranging from 32 to 16,384, and then resamples the channel to adjust the sample rate up or down by an arbitrary factor [14]. Moreover, each channel can produce complex data, namely I and Q components of the digitized signals. The DDC function can be configured in various modes, such as 16-Channel Narrow Band Configuration, 8-Channel Split-I/Q Configuration, and 4-Channel Wideband Configuration [14]. The 4-Channel Wideband Configuration is used for our development. The DDC output are stored in the FIFO as 2's complement 24-bit samples, which leads to the maximum absolute value is around 8.4×10^6.

Figure 2.6 shows the sampled I and Q components of sinusoidal input signals. Note that the data show the effectiveness of decimation functionality. Although

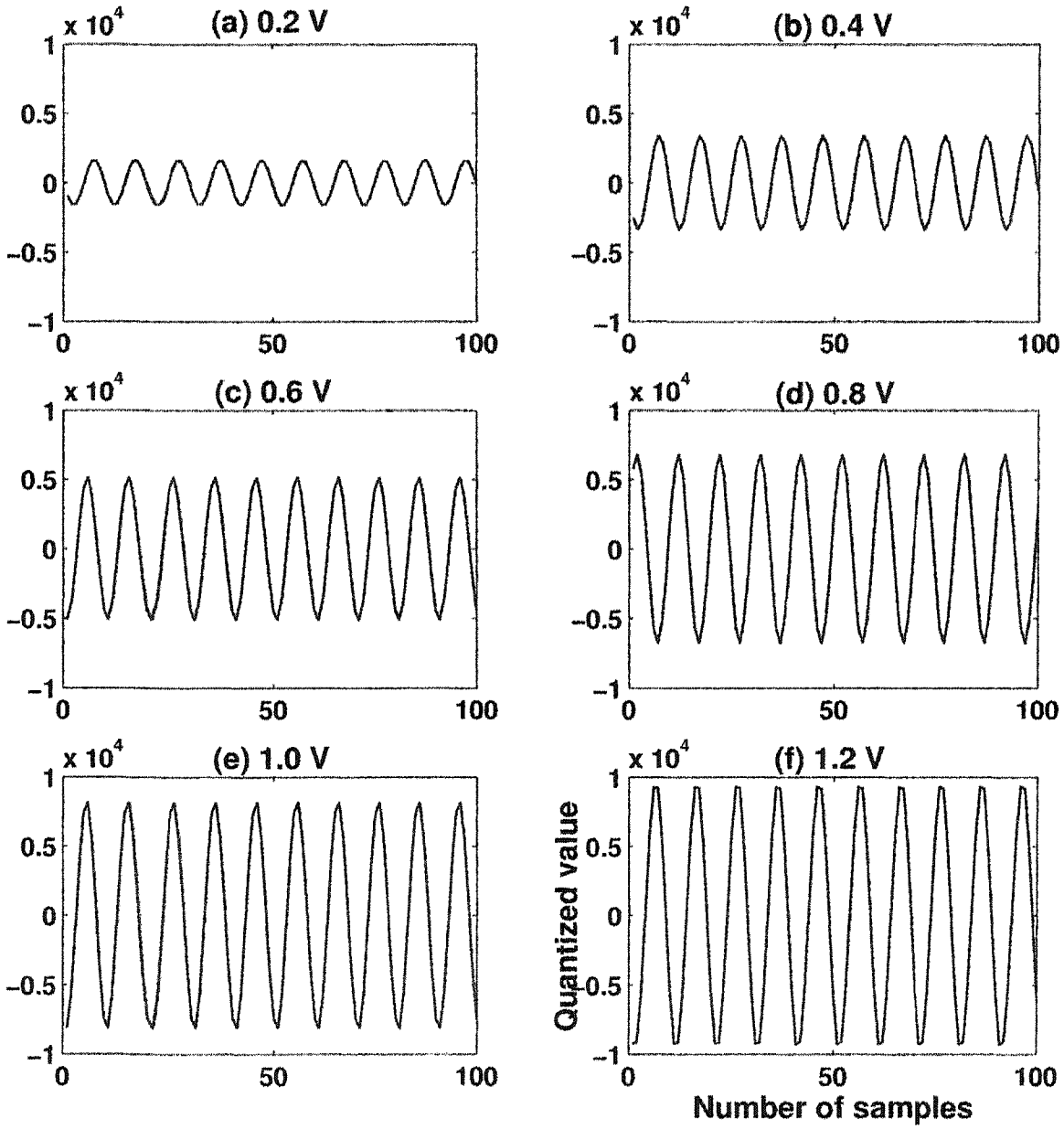


Figure 2.4: Input signals are continuous sine waves of the frequency of 10 MHz. The peak to peak voltage of the input signal varies from 0.2 V to 1.2 V. The sampling frequency is 100 MHz.

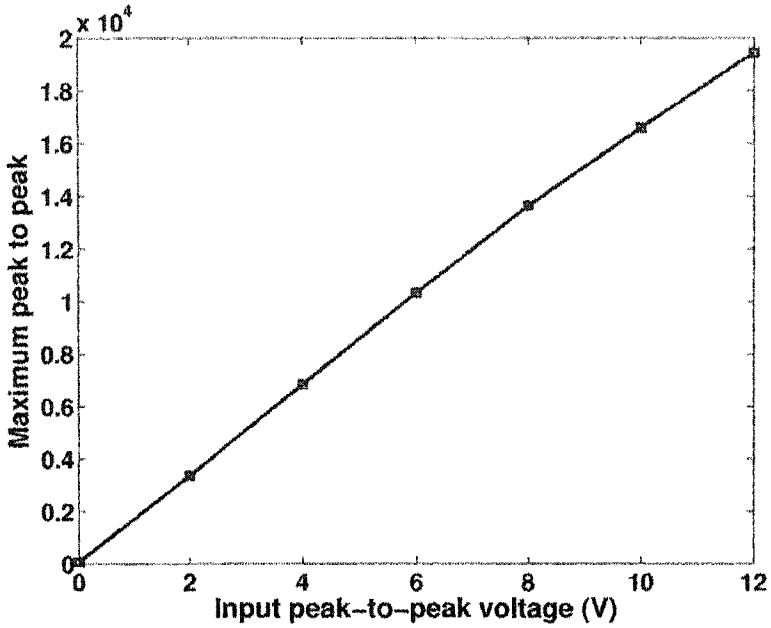


Figure 2.5: The quantized maximum peak-to-peak value which is proportional to the peak-to-peak voltage of input signal.

the data shown in Fig. 2.6 are quantized value at 1-MHz sampling rate after decimation, which is lower frequency than the input signal frequency of 10 MHz, they show clearly the sinusoidal characteristic of the input signal. Fig. 2.7 shows the linearity between the quantized output and the peak-to-peak voltage of the input signal in the DDC configuration too. In this measurement, the selected gains of the filters of each channel are listed in Table 2.1. The Cascade Integrate Combo (CIC) filter reduces the sample rate by a programmable factor [15]. The coarse gain circuit boosts the gain of weak signals. The Compensating FIR (CFIR) filter and the Programmable FIR (PFIR) filter, which contain a 21 tap and 63 tap respectively, decimate the output of the coarse gain circuit by two with programmable 16-bit coefficient.

Figure 2.8 shows the in-phase and quadrature components of the pulse-modulated sinusoidal input signal. The frequency of the sinusoidal signal is 10 MHz, and the pulse-modulation frequency is 1 KHz with duty cycle of 50%. The input signal was digitized at a sample clock of 40 MHz and was decimated by a factor of 40,

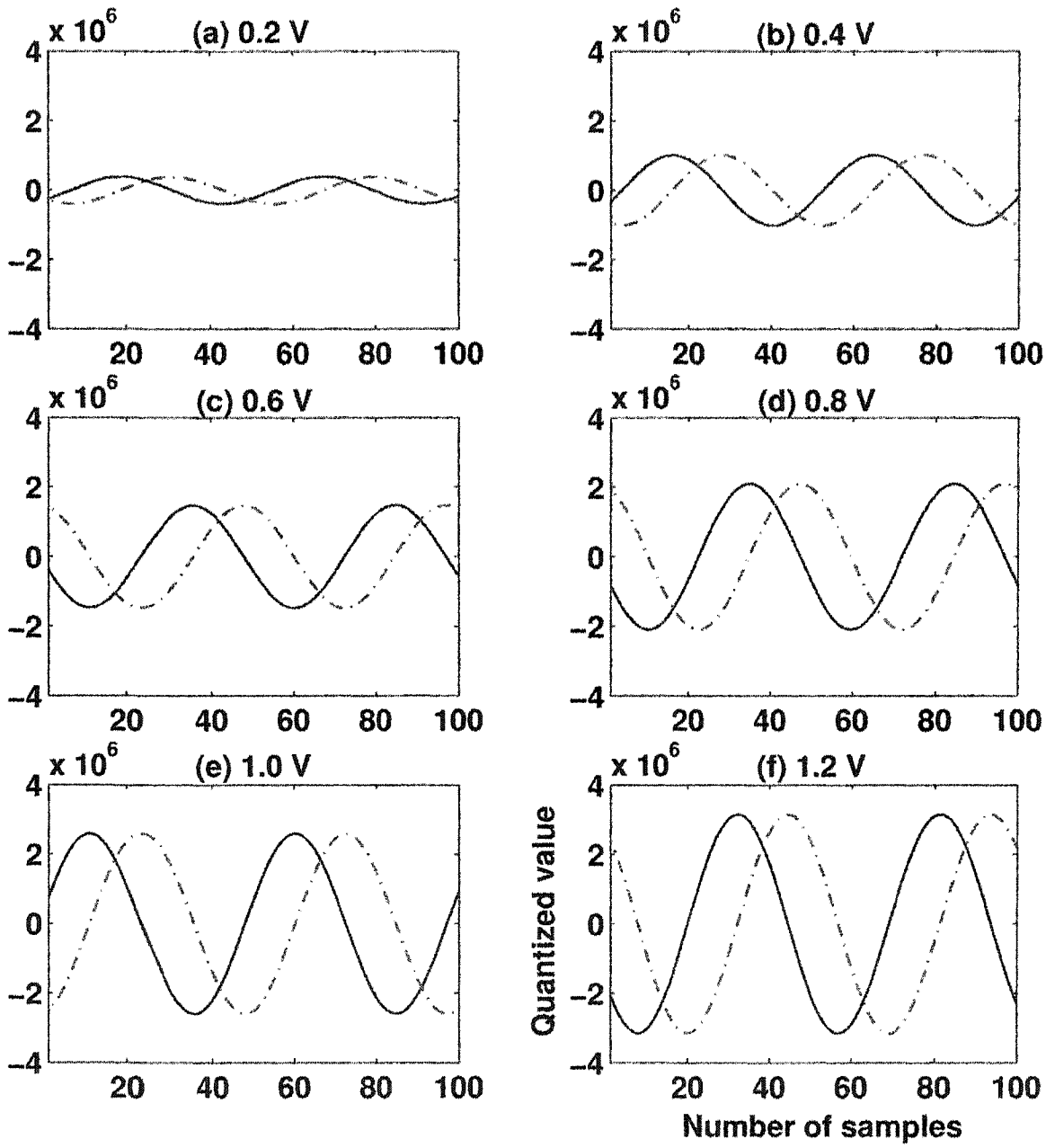


Figure 2.6: Input signal is a continuous sine wave with the frequency of 10 MHz. The peak-to-peak voltage varies from 0.2 V to 1.2 V. The sampling frequency is 40 MHz. Decimation factor is 40. The tuning frequency is 10 MHz.

Filter	Setting	Gain (dB)
CIC Filter	N=10	50
Coarse gain	off	0
CFIR Filter	CFIR_150	-0.7679
PFIR Filter	PFIR_150	-3.9857

Table 2.1: Selected gains of the filters of a down-converter channel contained in the GC4016 chip.

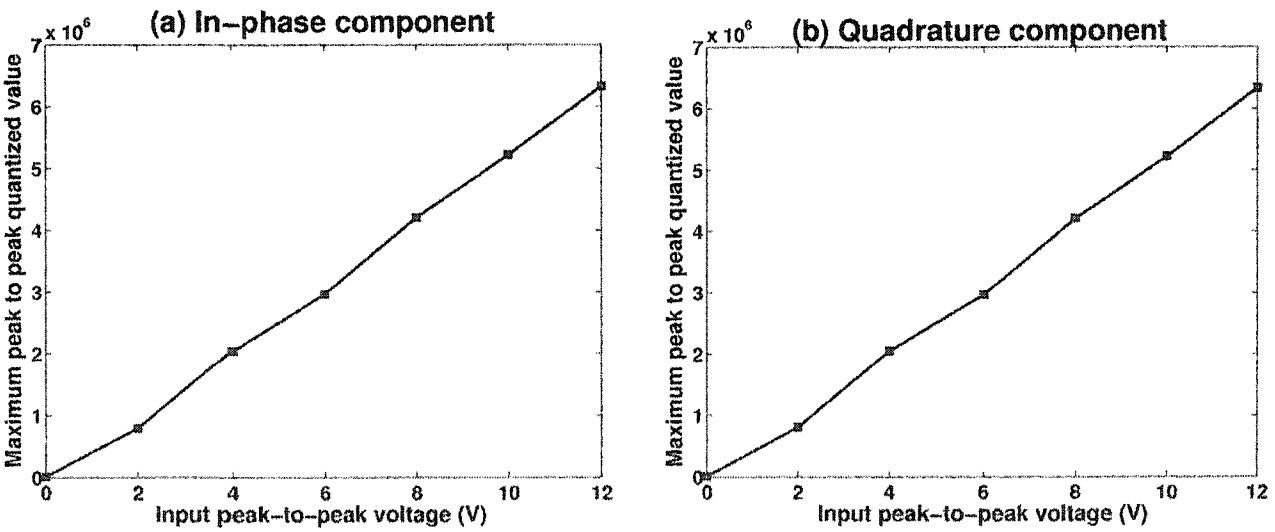


Figure 2.7: Change in the maximum peak to peak quantized value with the increase in the peak to peak voltage of input signal. (a) In-phase component and (b) Quadrature component.

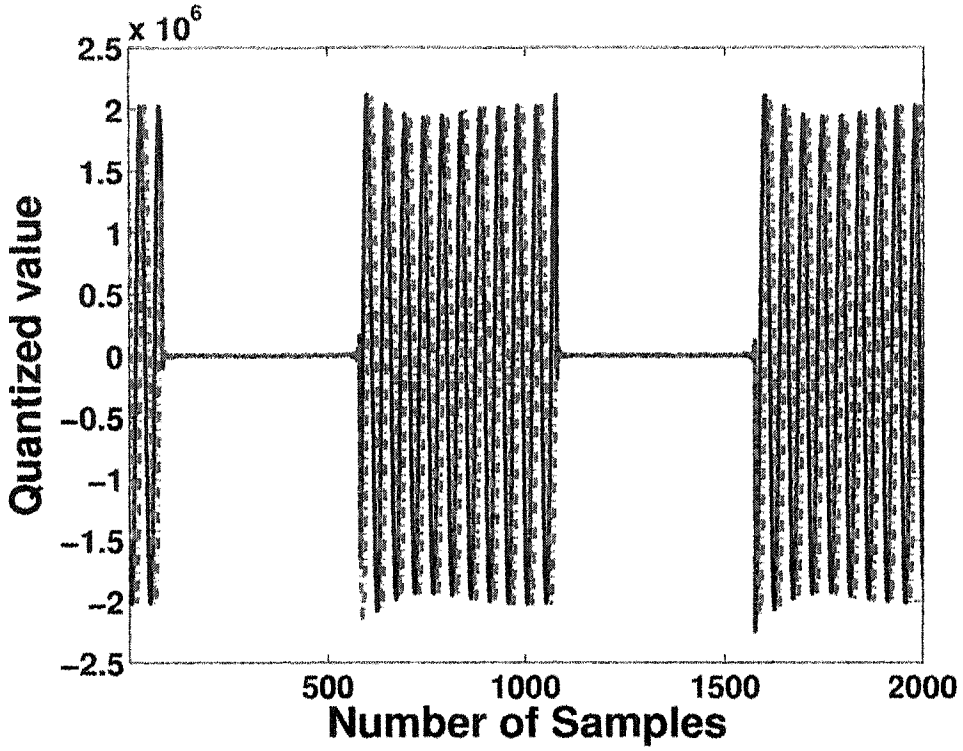


Figure 2.8: Quantized output of the pulse modulated (1 KHz) sinusoidal (10 MHz) input signal. The peak-to-peak voltage of the sinusoidal signal is 0.8 V. Sampling clock is 40 MHz and the sampled signal is decimated by the factor of 40. Tuning frequency is 10 MHz.

which produces the data output stream that is equivalent to a sampling rate of 1 MHz. Since great difference in the output can be found depending on the tuning frequency, the selection of the tuning frequency is important to get expected signal information from the DDC functionality. Here, another example of the effectiveness of decimation can be seen. During the one pulse cycle, 1000 samples are captured when the period is 1 msec and the sampling rate is 1 MHz. Moreover, ten sinusoidal waves are shown during on time of the pulse even though the frequency of the sinusoidal is 10 MHz.

Figure 2.9 shows the dynamic range of the ICS-554B-MN signal processor. The most significant 24 bits among 32 bits should have been shifted to right by 16 bits to produce 16-bit words, which is necessary for transmission to the remote

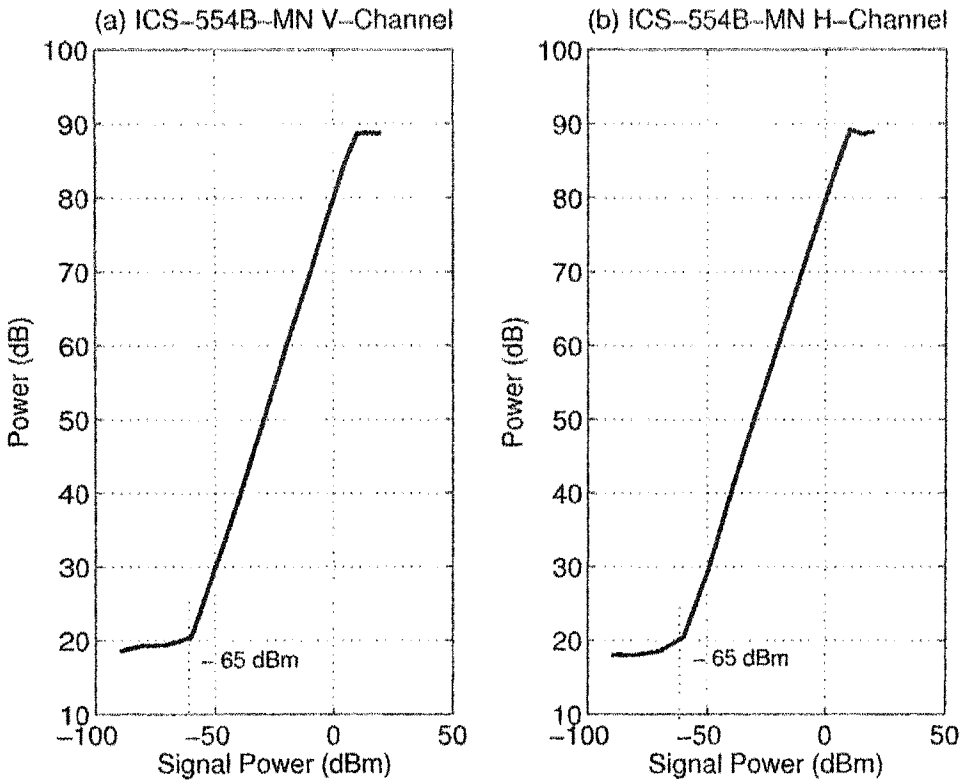


Figure 2.9: Dynamic range of the ICS-554B-MN ADC/DDC signal processor.

sites in order to save bandwidth. However, the data were shifted to right by only 14 bits to calibrate the power output from the parallel receiver to that from the DRX processor in evaluating the dynamic range and bandwidth. The dynamic range of the DRX processor is shown in Fig. 2.10 for comparison.

Figure 2.11 shows the bandwidth of the ICS-554B-MN card. It shows extremely sharp boundary in output power at round 10.5 MHz. The bandwidth of the DRX system is shown in Fig. 2.12 for comparison.

2.3.3 Acquisition of the DRS

Figure 2.13 presents the spectrum analyzer display of the down converted 10-MHz IF signals, in which the bandwidth is approximately 60 KHz. Figure 2.14 is an example of the received signals when the radar was pointing at the West, which shows some echoes bounded from Rocky mountains. The 10-MHz IF signals

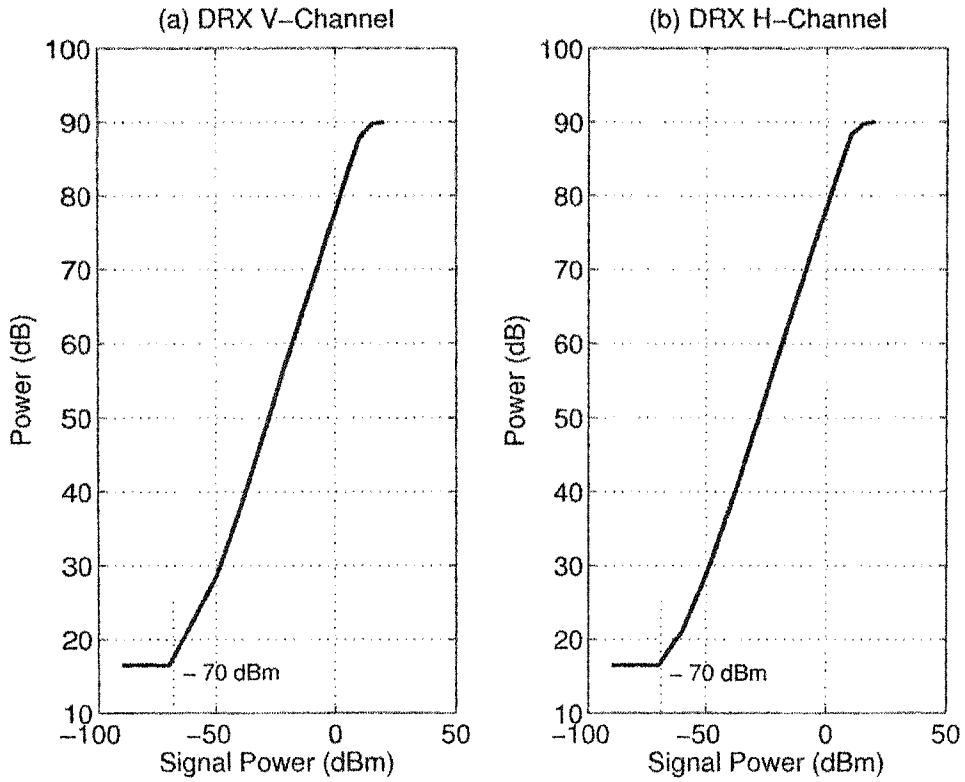


Figure 2.10: Dynamic range of the DRX signal digitizer/processor.

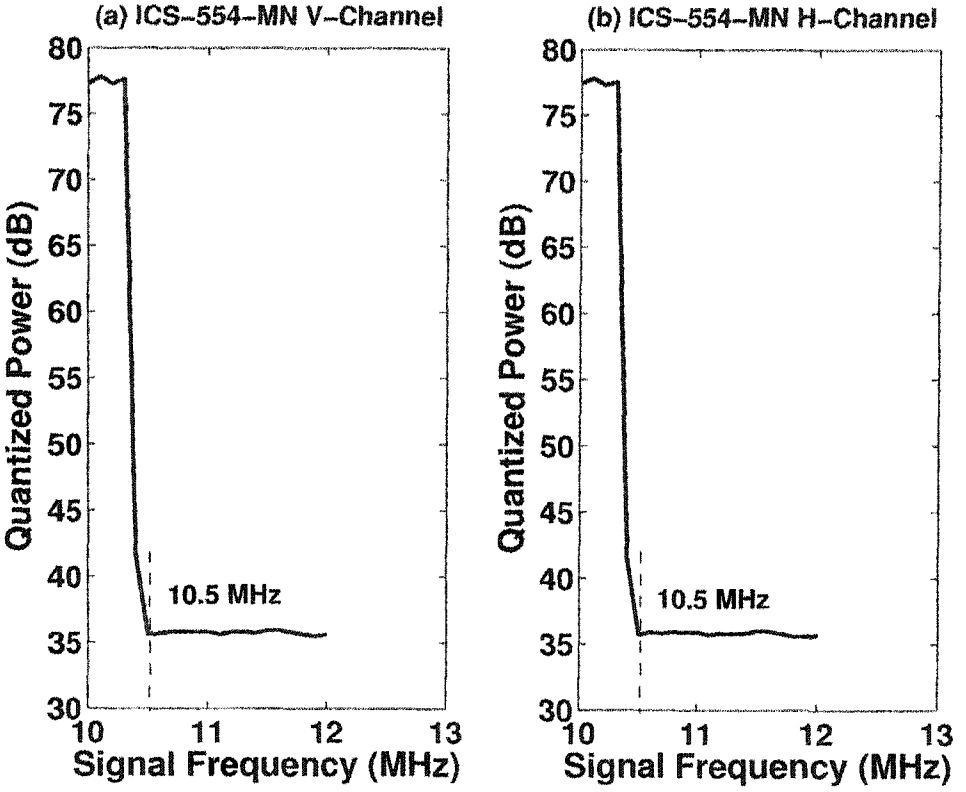


Figure 2.11: Bandwidth of the ICS-554B-MN ADC/DDC signal processor.

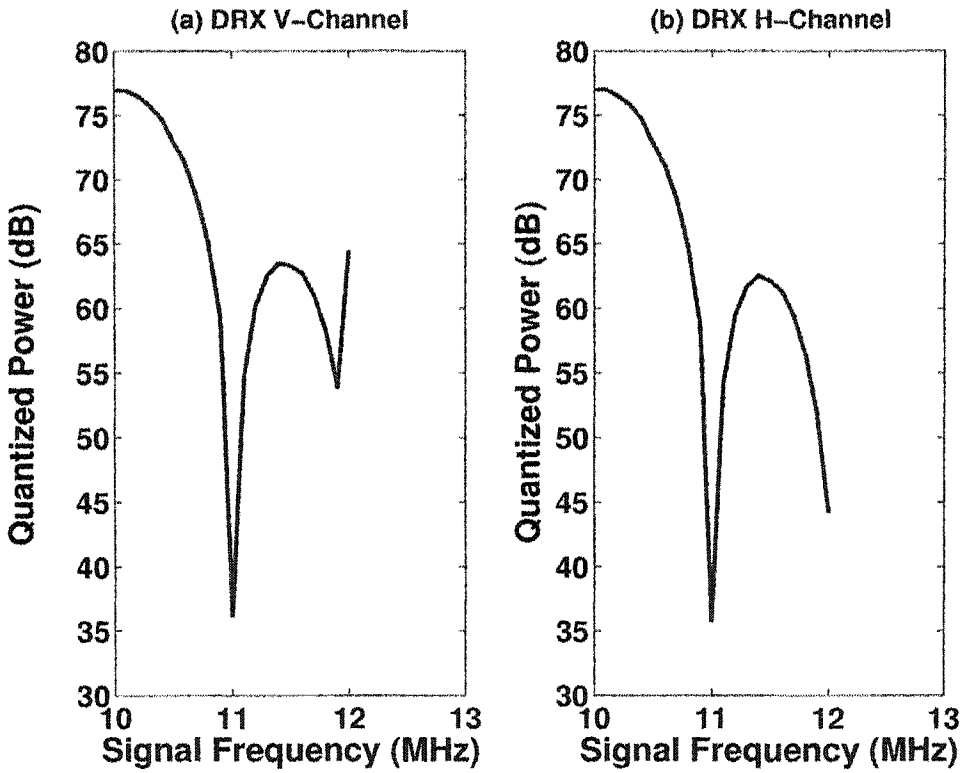


Figure 2.12: Bandwidth of the DRX signal digitizer/processor.

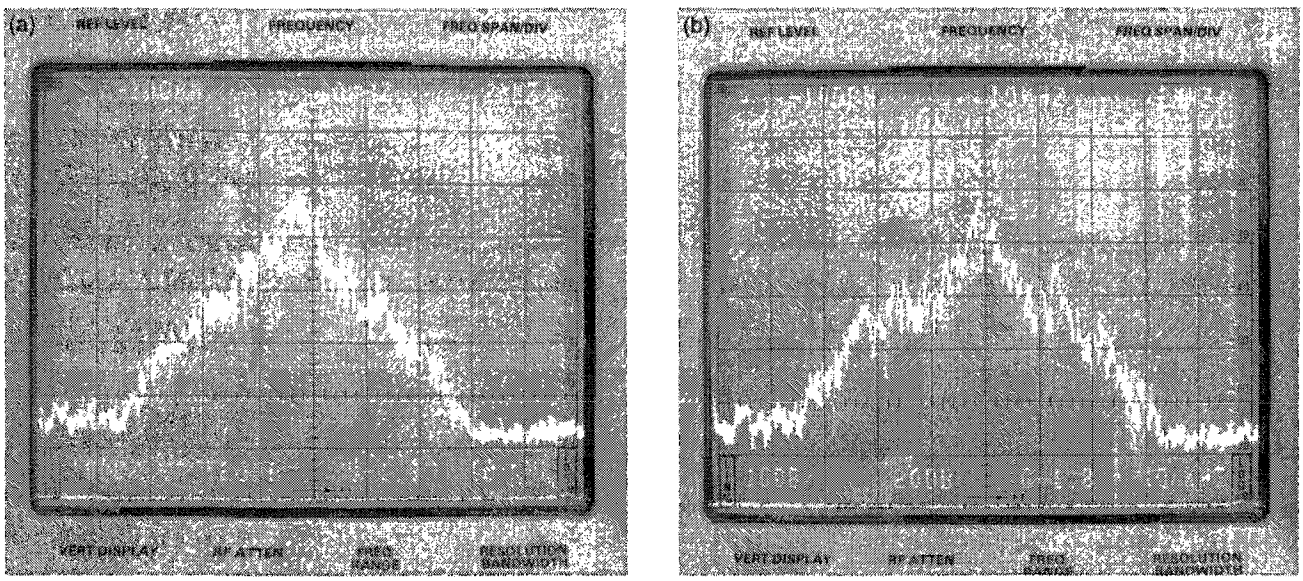


Figure 2.13: Spectrum analyzer display of the received signal that has been down converted to the 10-MHz IF. (a) Vertical channel and (b) Horizontal channel.

coming from the vertical and horizontal receivers are fed into the Channel 1 and the Channel 3 of the ADC/DDC signal processor, respectively. The signal processor starts to digitize the input signals into 14-bit words at the sampling clock of 40 MHz upon a receipt of the external sampling trigger signal that is synchronized with the pulse transmission. Next, the digitized data are tuned with the frequency of 10 MHz and extracted into I and Q components. Each component is filtered and decimated by the factor of 40, which leads to the data with the sampling rate of 1 MHz keeping characteristics of the received signals. The outputs of the vertical receiver and the horizontal receiver are stored in the FIFO 1 and FIFO 2 that are mounted on the ADC/DDC signal processor, respectively. Subsequently, the output data are delivered to the buffer of the DRS server from the FIFOs by the DMA mechanism. The ADC/DDC signal processor is set to be in Interrupt-Driven I/O mode such that when the data is ready to be read, the interrupt signal is invoked requesting the process to read the data from the buffer [16]. At last, the process reads the data and leaves the buffer empty.

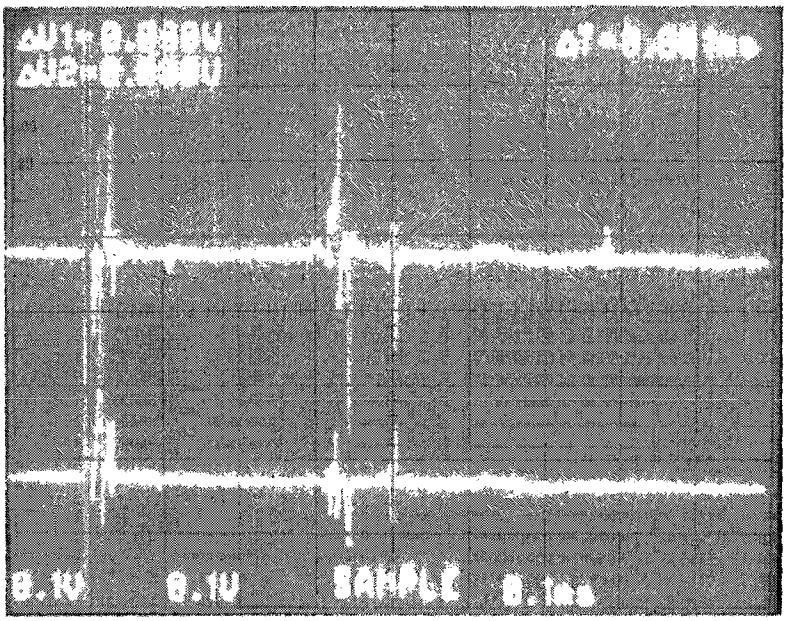


Figure 2.14: A snap shot display of the received signal. (Top) Vertical receiver and (Bottom) Horizontal receiver.

In principle, the pulsed-Doppler radar receives the back-scattered signal from targets after transmitting a pulse. Therefore, synchronization in sampling and digital-downconversion processing of the received signal with the pulse transmission is extremely critical to obtain expected result. The DRX processor generates the trigger signal when individual pulse is transmitted. The trigger signal is a rectangular waveform which originates from the modulation for the pulse transmission. Figure 2.15 presents the clock and the sampling trigger signal coming from the DRX processor. The time between two pulses of the trigger signal is identical with the PRT. The external sync generator produces a sync signal, which is applied to both external trigger and external sync input ports. Figure 2.16 shows the external sync signal and clock signal coming into the ICS-554B-MN signal processor. Here, the width of the sync should be approximately a period of the clock, which is critical requirement for suitable operation of the GC4016s. Since the external sync signal is fed into the trigger and sync ports, both the ADC and DDC are synchronized, which produces the range sample data sets from each

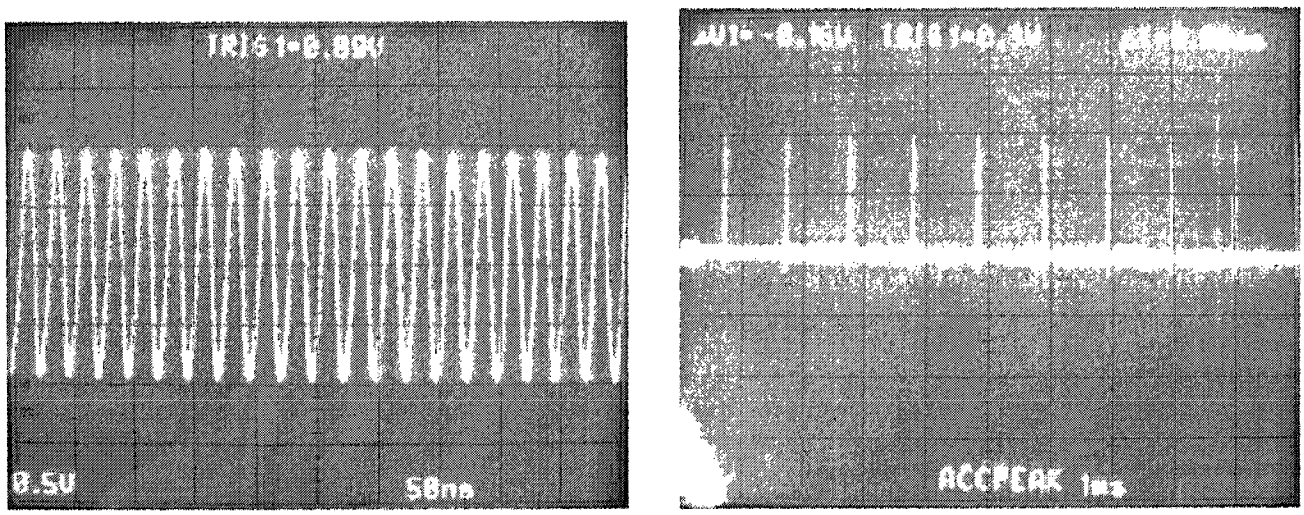


Figure 2.15: 40-MHz clock (Left) and Trigger signal synchronized with the pulse transmission (Right) generated by the DRX system.

pulse. Figure 2.17 shows internal connection of the GC4016 chips contained in the ICS-554B-MN signal processor. Not only four GC4016 chips but also four DDC channels contained in each GC4016 chip are synchronized with the external signal.

Figure 2.18 shows that the ADC is precisely triggered by the external sync. Figure 2.19 shows the DDC ouputs of the transmitted pulses produce jitters when the DDC is performed with the internal-sync mode. The internal sync is generated independently of the radar pulse transmission. Figure 2.20 shows the DDC outputs of the transmitted pulses when the DDC is triggered by the external sync. Extremely consistent DDC outputs are obtained with multiple pulse transmissions.

2.3.4 Acquisition of the Azimuth/Elevation angles and polarization states

The AZ/EL Interface circuit is placed between the position sensor and the digital I/O card as shown in Fig. 2.2. Figure 2.21 illustrates the circuit board and the detail circuit diagram can be found in Appendix B. The interface card converts the AZ/EL data in serial form coming from the position sensor to parallel data format, and keeps them in latches until they are read by the digital I/O card. The

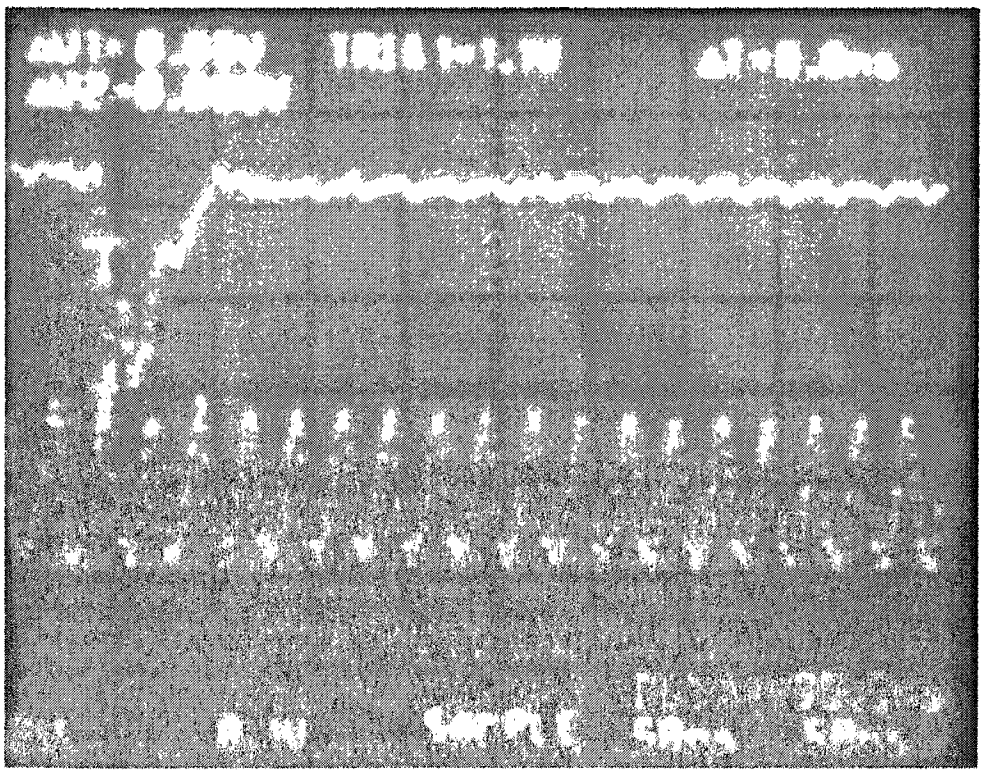


Figure 2.16: External sync and clock signals coming from the exteranl sync generator.

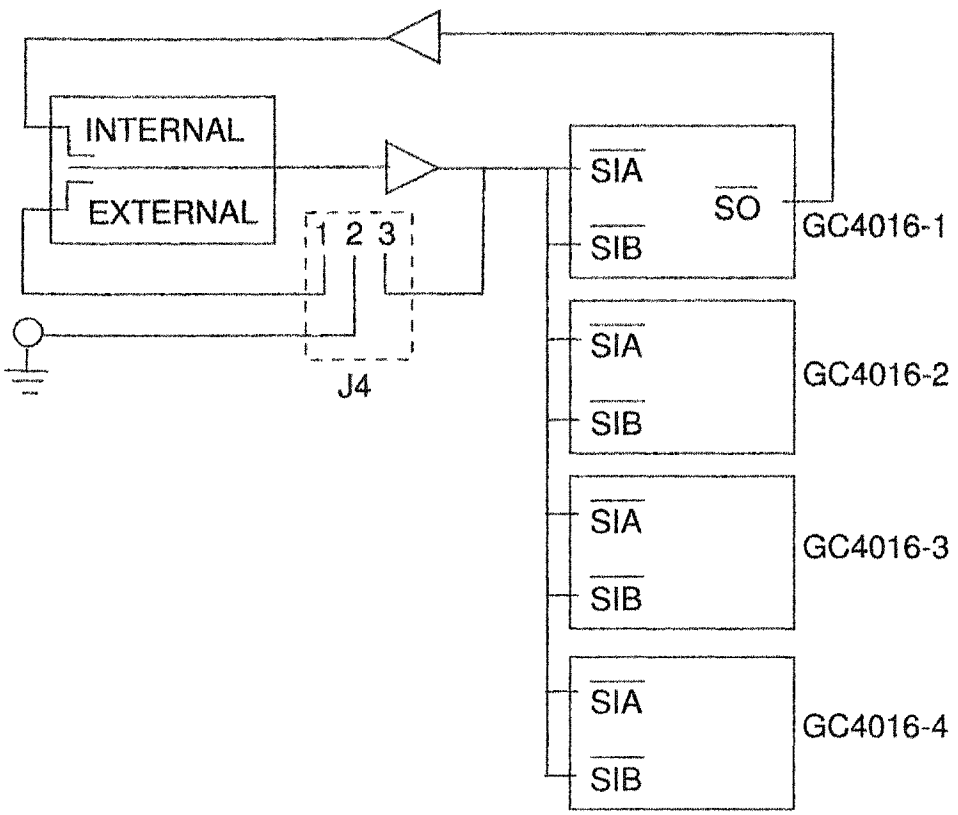


Figure 2.17: Schematic diagram showing internal configuration for the external synchronization of the GC4016 chips.

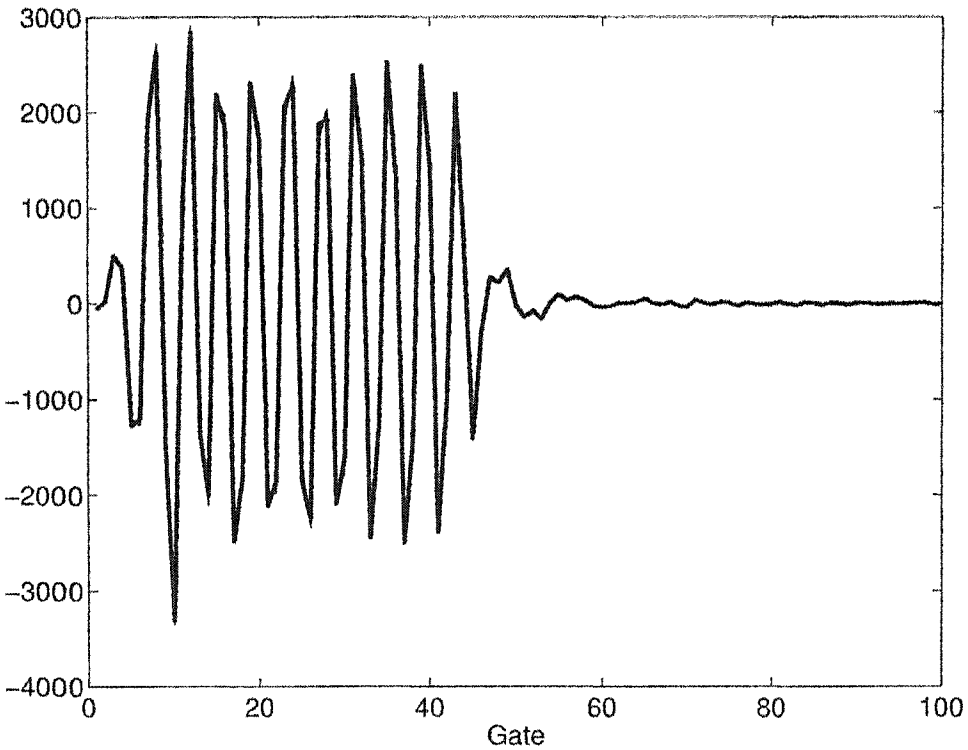


Figure 2.18: ADC outputs of the transmitted pulses of the CSU-CHILL radar. Five ADC outputs are overlaid. Sampling rate is 40 MHz.

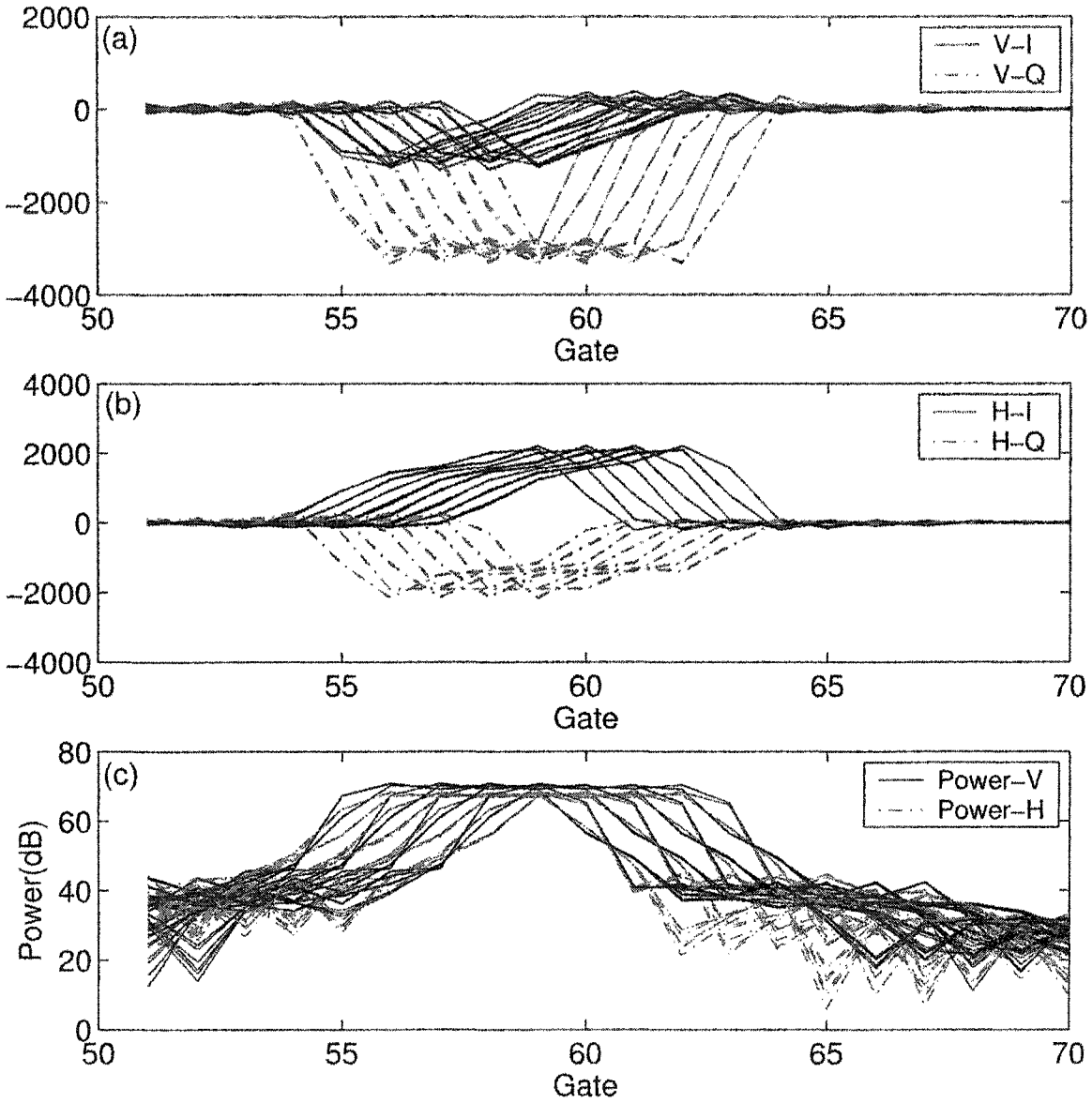


Figure 2.19: DDC outputs of the transmitted pulses of the CSU-CHILL radar when internally synchronized. The DDC outputs of fifty transmitted pulses are overlaid. Sample rate is 5 MHz. (a) vertical receiver, (b) horizontal receiver, and (c) power.

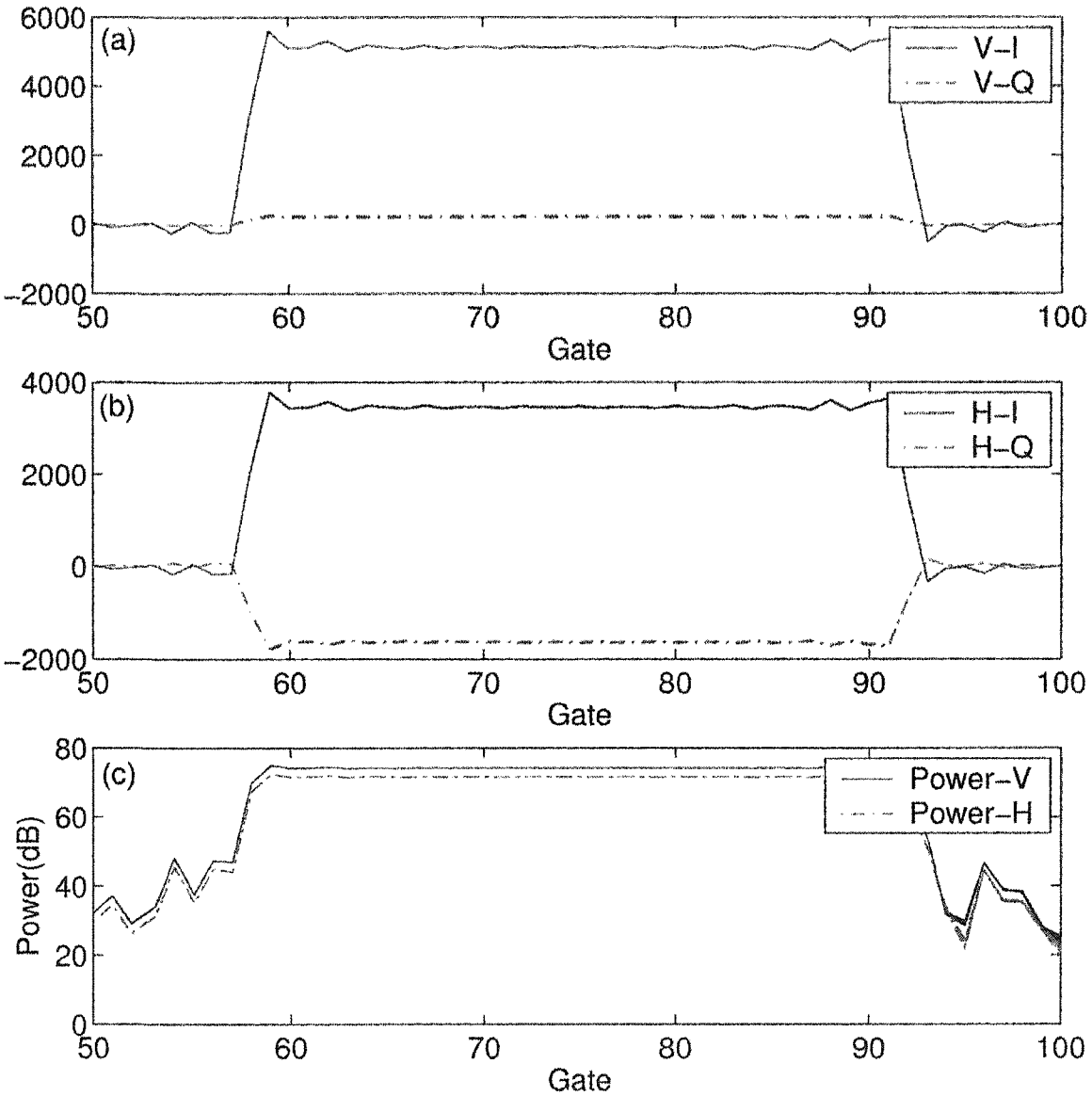


Figure 2.20: DDC outputs of the transmitted pulses of the CSU-CHILL radar when externally synchronized. The DDC outputs of fifty transmitted pulses are overlaid. Sample rate is 1 MHz. (a) vertical receiver, (b) horizontal receiver, and (c) power.

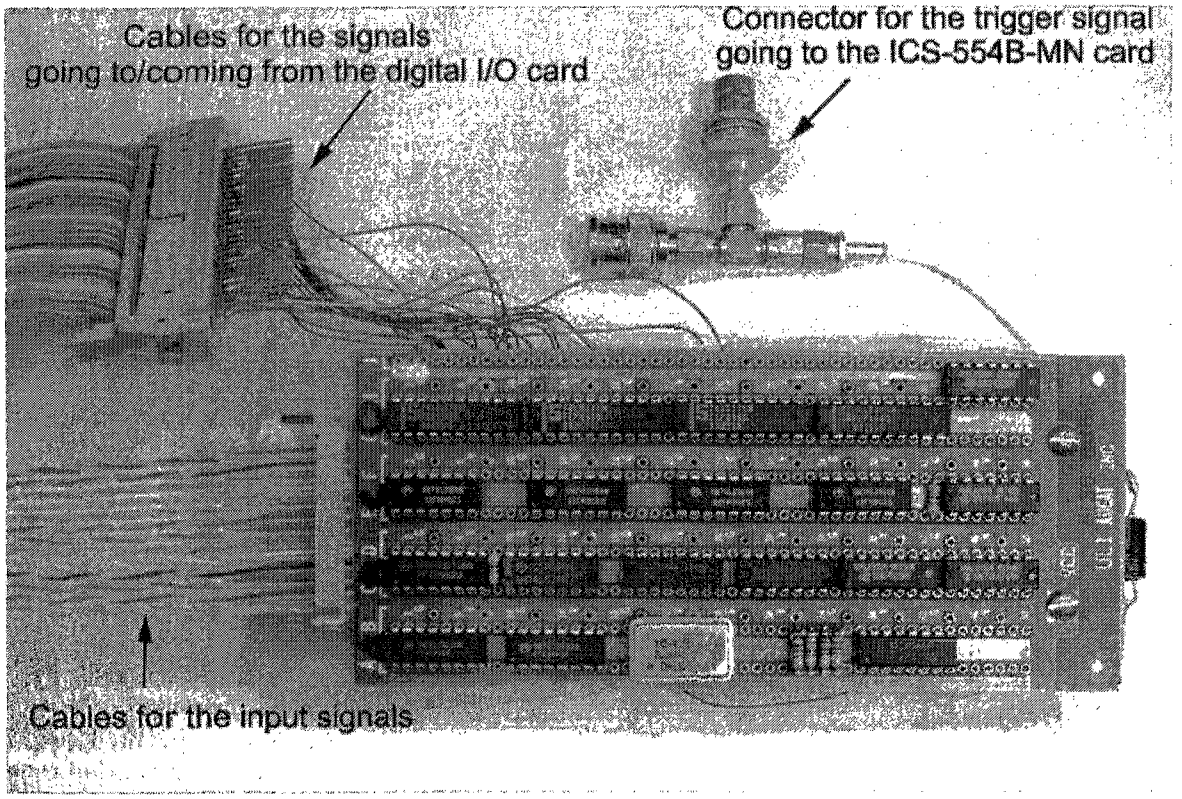


Figure 2.21: Azimuth/Elevation interface circuit.

trigger and polarization status signals are also amplified in the interface circuit and fed into the digital I/O card.

Two types of the trigger signals go into the digital I/O card as shown in Fig. 2.2. The sampling trigger signal is synchronized with the pulse transmission and lasts for 1 μsec, which also go into the ADC/DDC signal processor to start to digitize the received signals. Figure 2.22 (Left) shows the trigger signal coming out from the AZ/EL interface circuit, which indicates that the PRT is approximately 1 msec. Another trigger signal, which will be referred to as an integration cycle trigger, indicates the beginning of an integration cycle. Figure 2.22 (Right) shows the integration cycle trigger whose repetition time is approximately 128 msec because 128 pulses are transmitted with the PRT of $\sim$1 msec. It lasts approximately for a PRT as shown in Fig. 2.23 (Left). Comparion of the timing between the two trigger signals are shown in Fig. 2.23 (Right).

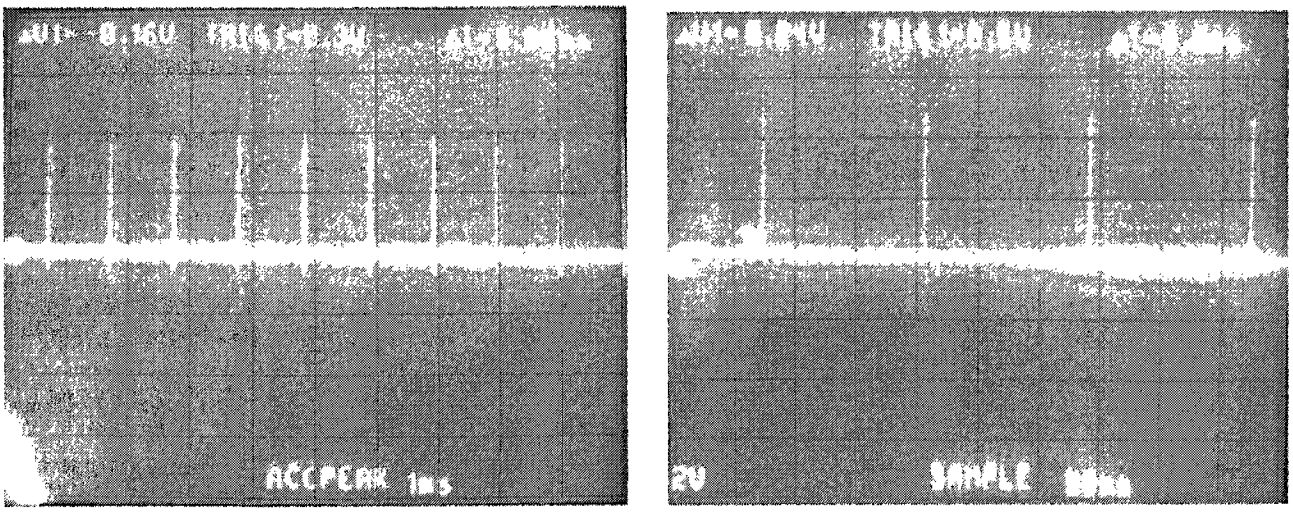


Figure 2.22: (Left) Sampling trigger signal and (Right) Integration cycle trigger signal.

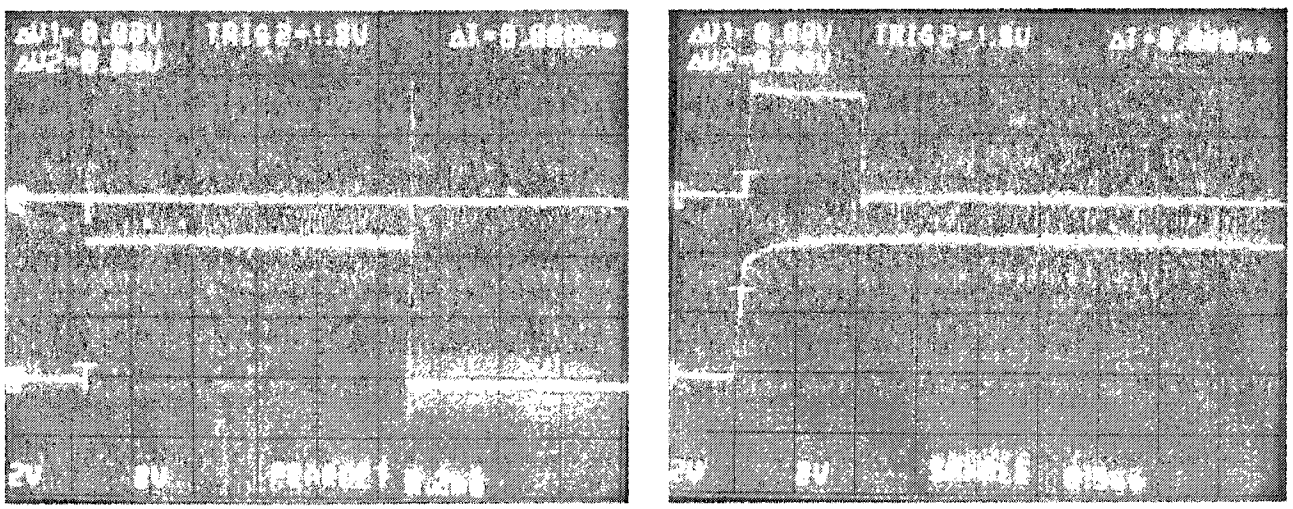


Figure 2.23: Comparison of the timing between the sampling trigger signal and the integration cycle trigger signal. (Left) time scale is 200 usec and (Right) time scale is 0.5 usec.

Digital I/O Ports	Input/Output	Signals
P1A0 - P1A7	Input	Lower 8 bits of AZ/EL
P1B0 - P1B7	Input	Higher 8 bits of AZ/EL
P1C0	Input	Trigger signal (Interrupt request)
P1C1	Input	Polarization status of the transmitted pulse
P1C2	Input	Integration cycle trigger
P1C4	Output	Selection of AZ (0) or EL (1)
P1C5	Output	Enable latches (1)

Table 2.2: List of the digital I/O ports that are connected to the AZ/EL interface circuit.

The digital I/O card (Model: PCI-7248) from the ADLINK is set to operate with an interrupt mode [17]. The sampling trigger signal is used as an interrupt request signal. Whenever the interrupt request is invoked, the interrupt signal handler routine, which is a part of the application program, is activated. The routine reads the polarization status of the transmitted pulse, and checks whether the integration cycle trigger is on. Figure 2.24 shows the signal that indicates the polarization states. The signal state lasts until the next pulse is transmitted. The process reads the Azimuth and elevation angles in series by sending a selection signal through a digital I/O port (P1C4) upon the receipt of the integration cycle trigger signal. After reading the Azimuth and elevation angles, the process sends another signal through a port (P1C5), which enables latches to keep a updated angle data for the next reading. Table 2.2 displays the list of connections between the digital I/O card and the signals coming from the AZ/EL Interface circuit.

2.3.5 Acquisition of the radar operating conditions

The radar operating parameters coming from the antenna controller are periodically delivered to the DRS server through the serial port. Each parameter is assigned with a specific value, and terminated by a semi-colon or newline to be distinguished each other. The parameters are classified into different items, such

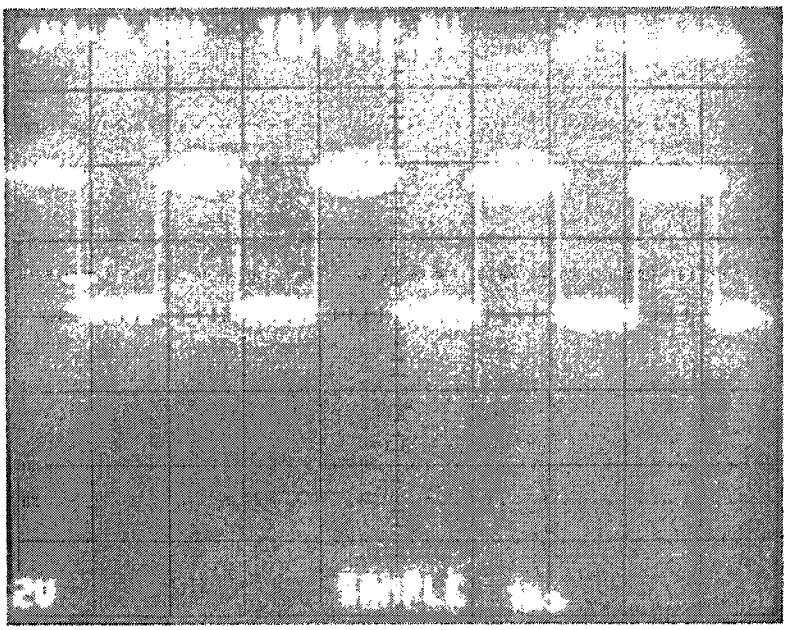


Figure 2.24: Signal indicating the polarization states of the transmitted pulses with the alternating mode. High and Low corresponds to the transmission with vertically and horizontally polarized states, respectively. For the hybrid mode operation, the application program does not concern about the polarization states.

as controlling the DRX processor operation, archiving DRX process activities, and informational items to be recorded. Among those parameter types, some of them that are associated with the fields of the ray header structure as shown in Table 2.3 are stored in a structure global variable in the application program.

2.4 Data acquisition process

The data acquisition process, which is a component of the DRS server, is an application program associated with the hardware systems described earlier. Functions of the data acquisition process includes initialization of the interfaced hardware systems, gathering of all data, arrangement of the data into a specified data format, and transfer of the data to a transmission process. The functions are divided into threads running simultaneously for the real-time operation. Main thread initializes the ADC/DDC signal processor and the digital I/O card, and prepares shared memory segments for passing the formatted data to the DRS

transmission process. It also writes the Azimuth and elevation angles and the polarization status of the transmitted pulses into global variables after reading them, and sets the flag to indicate the state of the integration cycle trigger signal.

The second thread reads not only the information on the radar operating conditions, but also the information necessary for the calibration of estimated parameters through the serial port. These information are periodically delivered to the DRS server and reserved in a global structure variable for the further reference. The global structure variable is repeatedly updated every time this thread gets new information.

The third thread reads the digitized I and Q components of the radar signal and arranges them into specified data formats. Table 2.3 and 2.4 show the structures of ray header and range sample header, respectively. When this thread detects an integration cycle trigger signal, it assembles a ray header by reading the global structure variable that keeps the updated information. For the ray header, the fields are classified into various categories, such as data identification, radar operating condition, calibration, and data transmission. These are necessary for the DRS client located at the remote sites to estimate the radar parameters. In addition, the thread assembles the range sample header before reading the output of the ADC/DDC signal processor from the buffer mounted on the DRS server. The information included in the range sample header helps the DRS client to identify where the received range sample data set belongs to. For a range sample data set, the range sample header is reserved in the front and followed by the series of the gate data.

Depending on the radar operating mode, the attribute of the arranged data set changes. For each gate, the I and Q components of vertical channel are located in lower number of address, and the I and Q components of horizontal channel are stored in the next address. For the alternating mode, all the gates have the

Data Identification	
Header ID	0 - ray header, 1 - range sample data set header
Radar ID	Unique number for a specific radar
Start time	Unix Date and time in word
Radar operating condition	
OP mode	0 - V only, 1 - H only, 2 - VH, 3 - VHS
Scan mode	0 - RHI mode, 1 - PPI mode
Volume number	Integer
Sweep number	Integer
Ray number	Integer
Azimuth	Azimuth angle degrees × 1000000
Elevation	Elevation angle degrees × 1000000
Pfr	Pulse repetition frequency × 1000
Ngates	Number of gates
Gate spacing	Millimeters
Start range	Millimeters
Npulse	Number of pulses transmitted at a particular angle
Calibration	
Txmit power H	Horizontal peak transmit power dBm × 100
Txmit power V	Vertical peak transmit power dBm × 100
Receiver gain H	dB × 100
Receiver gain V	dB × 100
Zdr offset	dB × 1000
Noise power H	dB × 1000
Noise power V	dB × 1000
Phidp rotation	degree × 1000000
Test type	0 - no test, 1 - Zdr calibration others TBD
Transmission	
Ndata set	number of range sample data sets in a data packet
RTT	round trip time(msec)
Transfer rate level	1 - 10
Transport protocol	0 - TCP, 1 - UDP

Table 2.3: Structure of ray header that consists of 28 fields. Data type is integer and each field occupies 4 bytes.

Header ID	0 - ray header, 1 - range sample data set header
Volume number	Integer
Sweep number	Integer
Ray number	Integer
Data number	Sequence number of range sample data sets in a ray
Polarization	0 - Vertical radiation, 1 - Horizontal radiation
Data code	0 - normal data, 1 - last data in a ray,
	2 - retransmitted data

Table 2.4: Structure of a range sample header that consists of 7 fields. Data type is integer and each field occupies 4 bytes.

received signals for pulse 1 (V transmission), i.e. a copolar sample (I_{vv}, Q_{vv}) and cross-polar sample (I_{hv}, Q_{hv}), followed by the received signal of all the gates for pulse 2 (H transmission), i.e. a cross-polar sample (I_{vh}, Q_{vh}) and copolar sample $(I_{hh}, Q_{hh})$and so on. In the hybrid mode, all the gates have the received signal for each pulse, i.e. (I_{vv}, Q_{vv}) and (I_{hh}, Q_{hh}) and so on. In latter mode, there are no cross-polar received signals. These ray header and the range sample data set are written into a shared memory segment and passed to the DRS transmission process. Detail description on the whole end-system architectures for the real-time operation of the high-bandwidth VCHILL will be provided in the next chapter.

2.5 Implementation and Performance Evaluation

2.5.1 Critical factors for implementation

The new parallel receiver has been implemented on the standard PC containing Xeon dual processors (2.8 GHz). Operating systme is Red Hat 8.0 Linux. The ADC/DDC signal processor and the digital I/O card are interfaced with a PCI-X (64-bit/66 MHz) and a PCI (32-bit/33 MHz) slots, respectively. Fig. 2.21 illustrates the AZ/EL interfacing circuit made by hand. The application program was implemented using C language, System V IPC and Posix threads with gcc

compiler. In addition, Application Programmers Interfaces, which are provided by the manufacturers to access the ADC/DDC signal processor and the digital I/O card [18, 19], were used in coding.

The transfer rate from the FIFO to the buffer mounted on the DRS server should be fast enough not to lose any data stored in the FIFO because of overflow. The transfer rate for a single channel is 128 Mbps when the sampling rate is 1 MHz. Here, the DDC data output format of the signal processor includes the data generated by the unoccupied input channel 2 and 4 by default, which consume the half of the transfer bandwidth [14]. The maximum bandwidth of the PCI local bus is theoretically 4.224 Gbps for the 64-bit/66 MHz PCI-X slot. This bandwidth is equivalent to the sampling rate of 16.5 MHz. However, since the highest sampling clock is 100 MHz and the lowest decimation factor is 8 according to the specification of the signal processor [14], the highest equivalent baseband sampling rate is 12.5 MHz. In our development, the highest sampling clock comping from the DRX processor can be doubled up to 80 MHz. Therefore, the maximum sampling rate achievable is 10 MHz.

The computer system should be set to maximize CPU performance. For example, the enabled hyper-threading option can greatly increase the CPU performance, such as reading speed from the buffer to the application program, which eventually increases the transfer rate from the FIFOs to the buffer [20]. If the transfer rate is lower than the data rate, some stored data in the FIFO are lost arbitrarily during operation because of the overflow, which leads to the failure in providing expected outputs. It was experienced that when the DRS server was booted with disabled hyper-threading, user display showed arbitrary range offsets from the center due to the unpredictable lost of output data stored in the FIFO.

The size of a FIFO is 4.608 MB (72K $\times$ 64) and each gate data set of a single channel (I and Q components) occupies 16 bytes. Therefore, a FIFO can collect

up to 288K gates data. As mentioned earlier, half amount of the gate data are dummy coming from the unoccupied channels. When the number of gates is 1000 and 128 pulses are transmitted at a pointing angle, a FIFO can store up to data amount of two and one fourth rays. The time taken to fill up a FIFO is around 288 msec assuming the PRT is 1 msec. Even though the transfer rate is faster than the data rate, the server might not be able to read the data from the FIFO in the beginning of operation because of initialization procedures that last longer than the time taken to fill up the FIFO, leading to some loss of outputs. This also makes arbitrary range offsets varying every time the system operates. To remove this problem, the implementing code should complete lengthy initialization procedures before enabling the ADC/DDC signal processor.

2.5.2 Calibration of the range position

RF Test signal, which can be applied between the antenna and the receiver, has been used to calibrate the parallel receiver system. Figure 2.25 presents an example of the ascope display showing the test signal. The equivalent number of gates along which the test signal is applied is approximately 30. The test signal was used for calibration of the absolute gate position compared to the DRX system.

Figure 2.26 shows the unexpected reflectivity along the gates for the three consecutive rays in early stage of implemenation with the applied test signal. Long plateau of high reflectivity, which lasts along the gates in a cyclic fashion, is shown for each ray instead of a spike of reflectivity. The number of gates that make the plateau was counted to be approximately 512. Figure 2.27(a) illustates the origin of the problem. Because of the feature of the ADC/DDC signal processor, it produced additional output of a frame that are equivalent to four gate data. When the amount of data to be read by the processor was set based on 1000 gates, the dummy data were continuously accumulated in the FIFO. Consequently, the

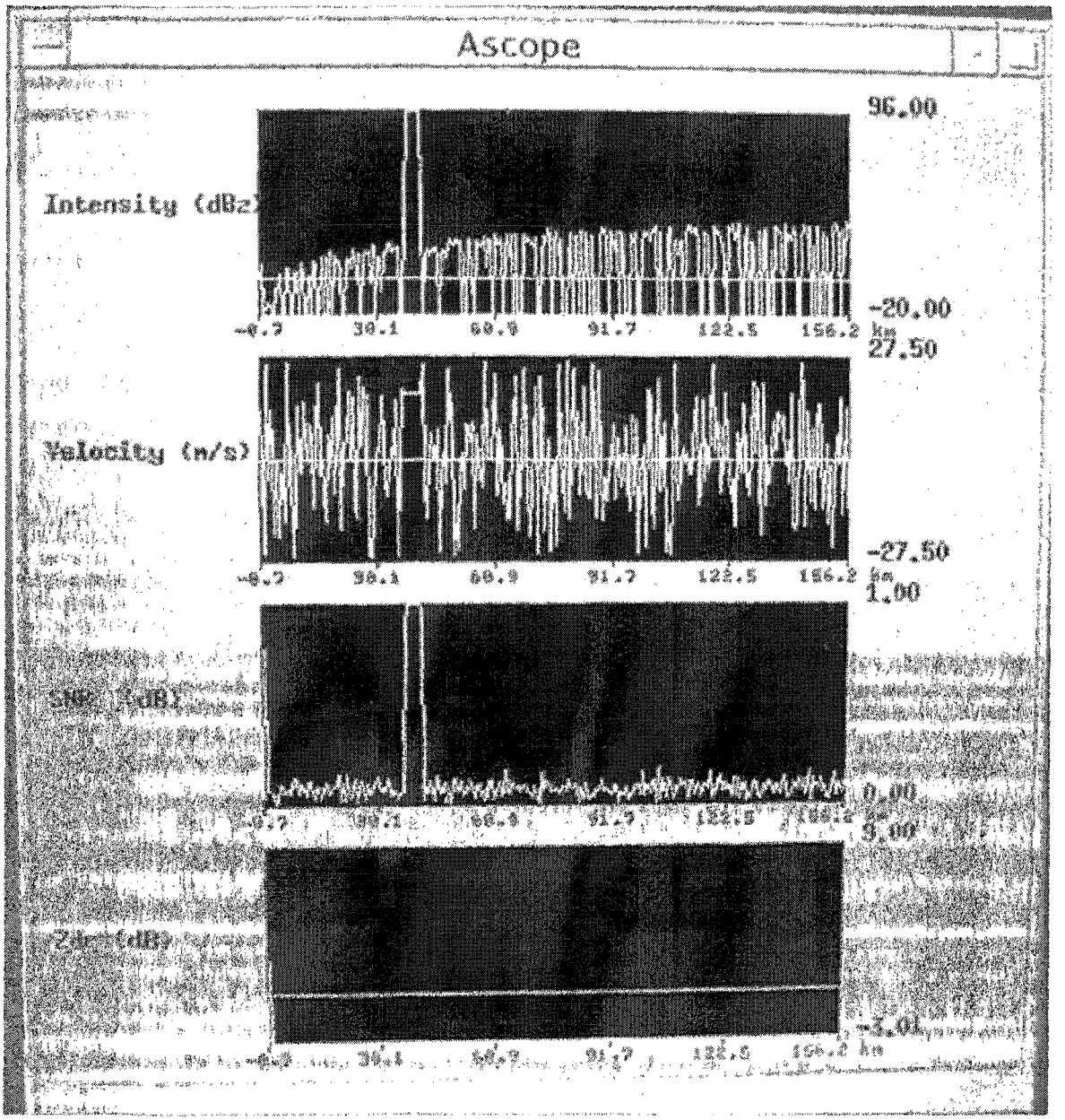


Figure 2.25: An display of ascope showing test signal that are applied to the receivers. The ascope continuously displays the radar parameters coming from the DRX processor along the gates.

relative position of the gates along which the test pulse was applied was shifted to outward ranges. Therefore, when the parameters were estimated with the 128 range sample data sets, the reflectivity plateau with the total shifted number of gates of 512 was generated as shown in Fig. 2.26. To solve the problem, we incremented intentionally the amount of data to be read by four gate as shown in Fig. 2.27(b). Users can change the settings of the DDC module configuration which determines the amount of a range sample data set to be read by the signal processor. Figure 2.28 shows the spike of reflectivity displaying the test signal as expected.

Absolute position of the gate was also calibrated compared to the DRX system. After solving the previous problem, larger range offset by 56 gates compared to the DRX system was found because of the intrinsic system setting of the ADC/DDC signal processor. In order to get rid of the offset, the additional 56 gates data are discarded at the DRS server just after being read from the buffer. Figure 2.29 shows the comparison of displays produced from two different receiver systems.

2.5.3 Data comparison with the DRX system

To demonstrate the functionality of the parallel receiver, the radar parameters are compared between the parallel receiver and the DRX receiver as shown in Fig. 2.30 - Fig. 2.34. The radar was pointing at a specific angle that reflected strong echo signals from precipitation during received signal acquisition. The parameters of each gate were obtained by averaging the estimated parameters of the gate over 50 rays for both the parallel receiver and the DRX receiver. Mainly the parameters are compared along the gates which were in precipitation. The parameters obtained from the parallel receiver show good coincidence with those obtained from the DRX receiver even though minor calibration is required. The comparison demonstrates that the parallel receiver operates properly for the high-bandwidth VCHILL.

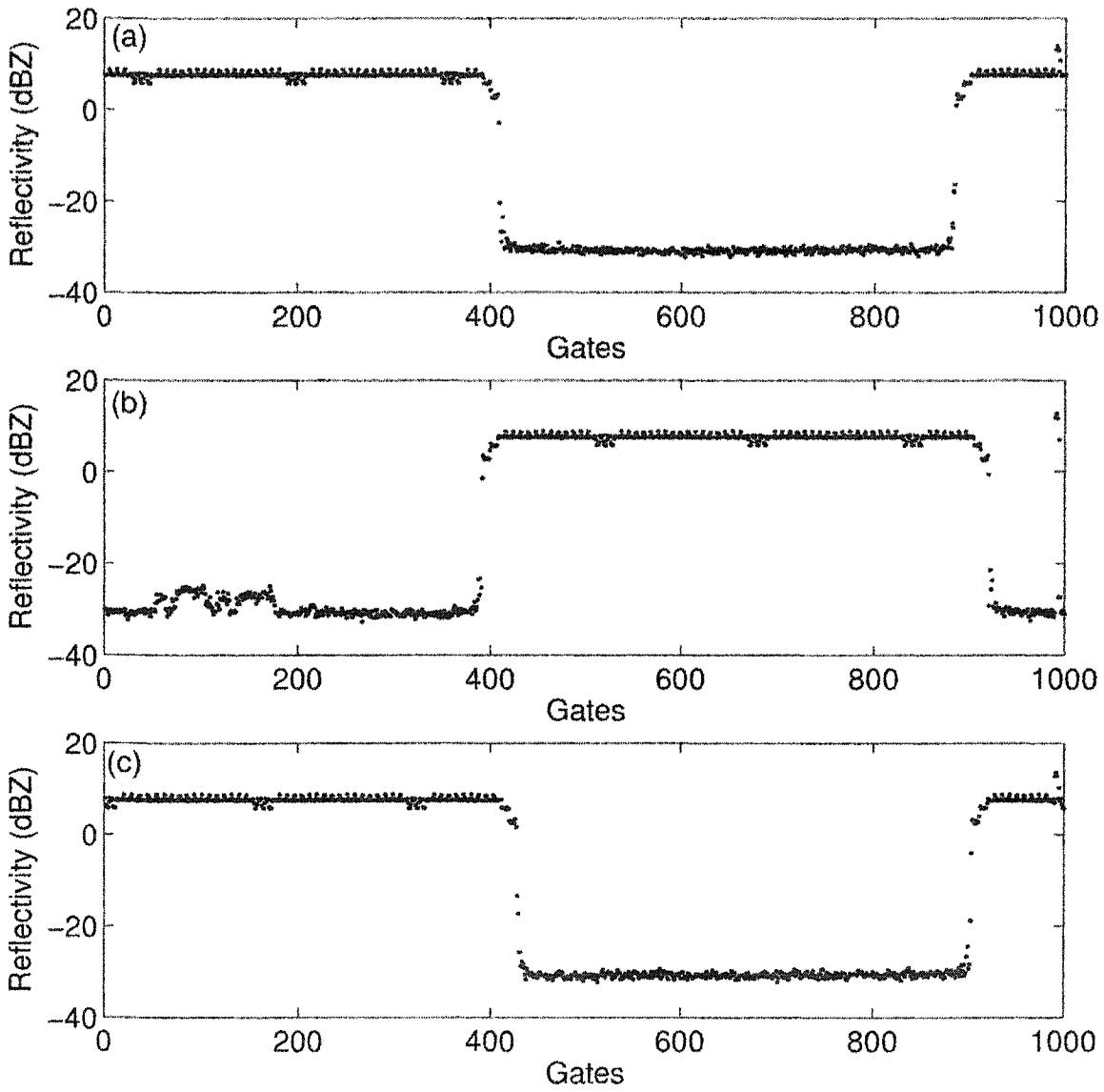
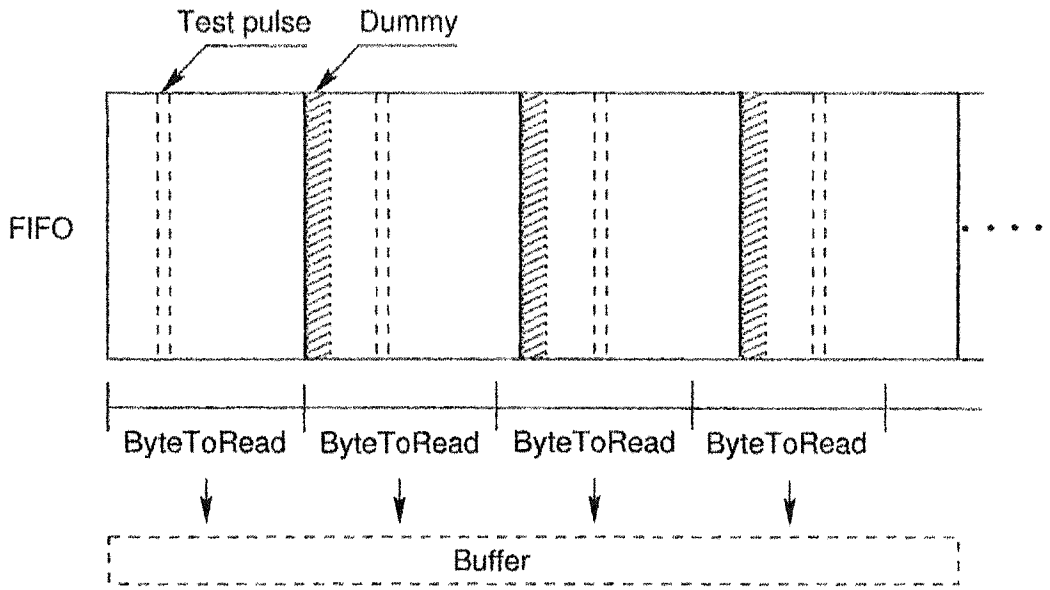
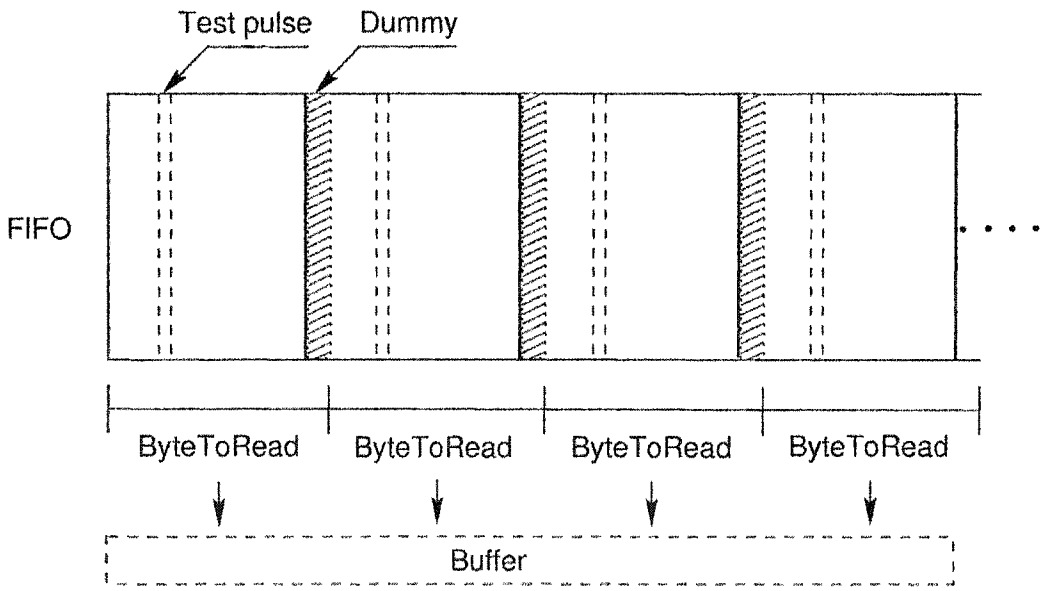


Figure 2.26: Reflectivity of the three consecutive rays along the gates. (a) ray number 59, (b) ray number 60, and (c) ray number 61. The reflectivity was not calibrated yet when they were measured.



(a)



(b)

Figure 2.27: Illustration showing the output of the ADC/DDC signal processor stored in the FIFO. The ADC/DDC signal processor creates additional data that are equivalent to four gates. The amount of data to be read by the processor was set (a) based on the operating manual, (b) by the intentional increment by four samples.

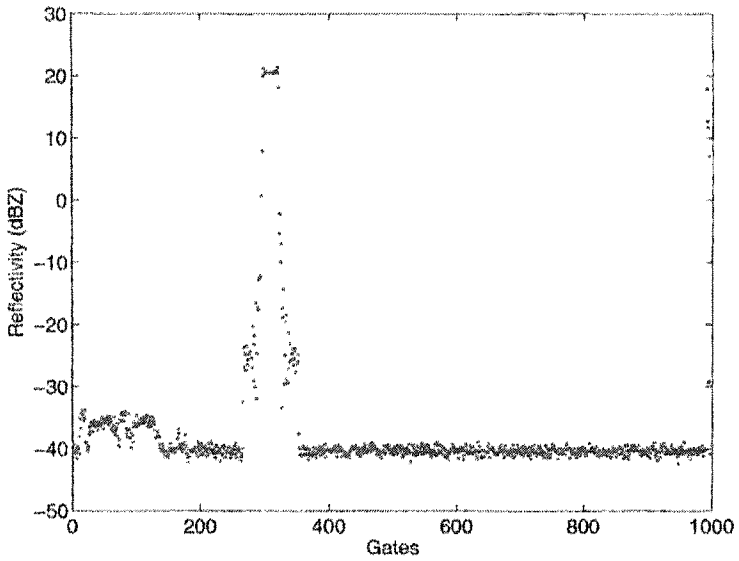


Figure 2.28: Spike of the reflectivity representing the test signal applied to the receiver. The reflectivity was not calibrated yet when it was measured.

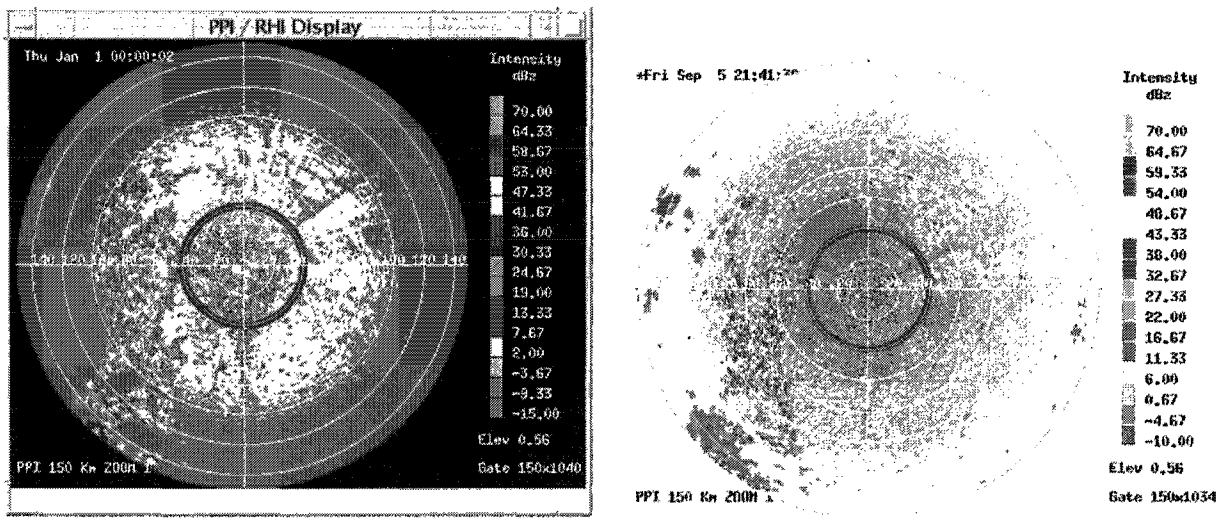


Figure 2.29: Comparison of the displays between the parallel receiver and the DRX system. Rings due to the test pulse are shown in the same ranges.

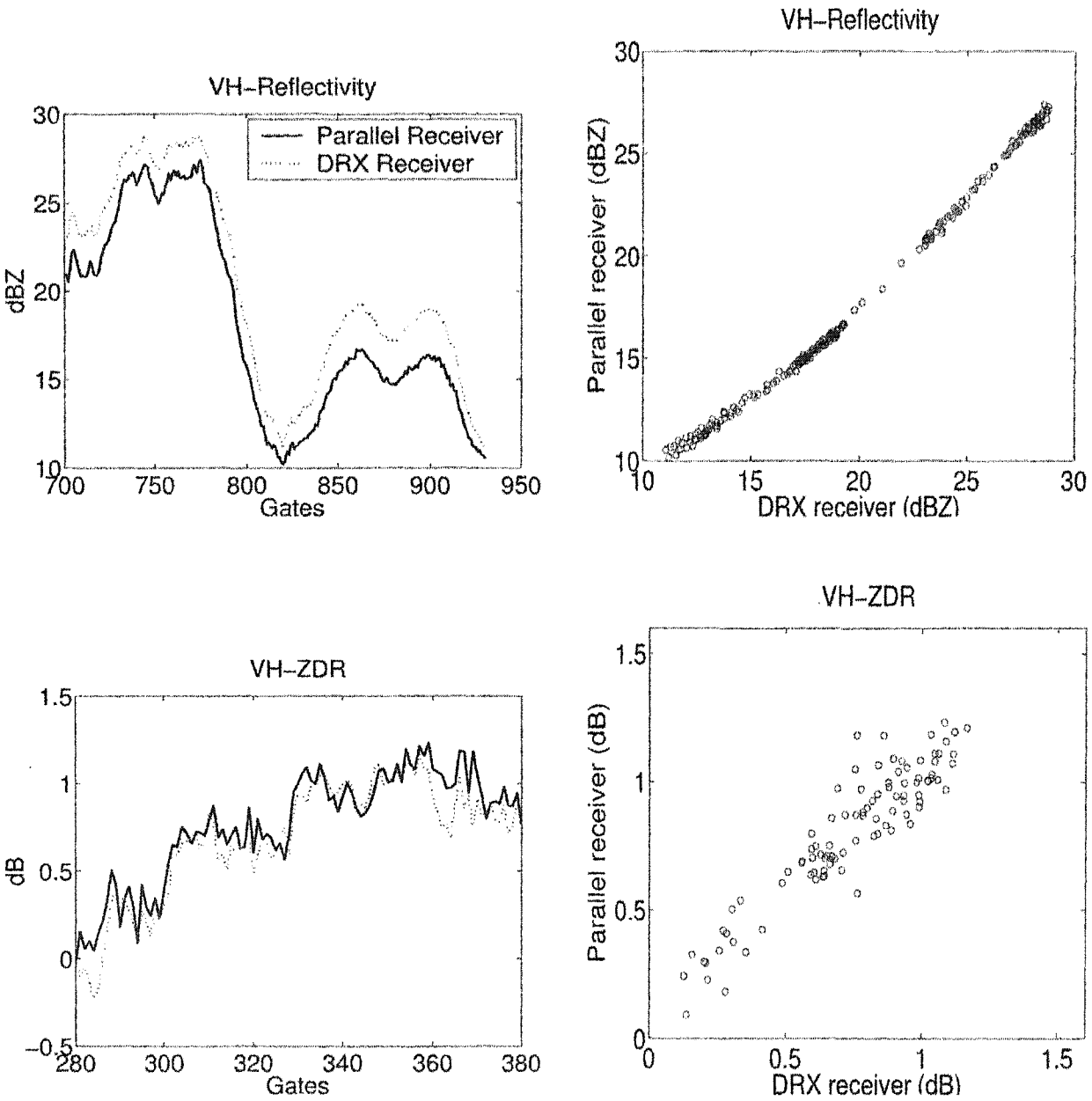


Figure 2.30: Reflectivity and Z_{dr} are compared between the parallel receiver and the DRX receiver. CSU-CHILL radar was operated in alternating mode.

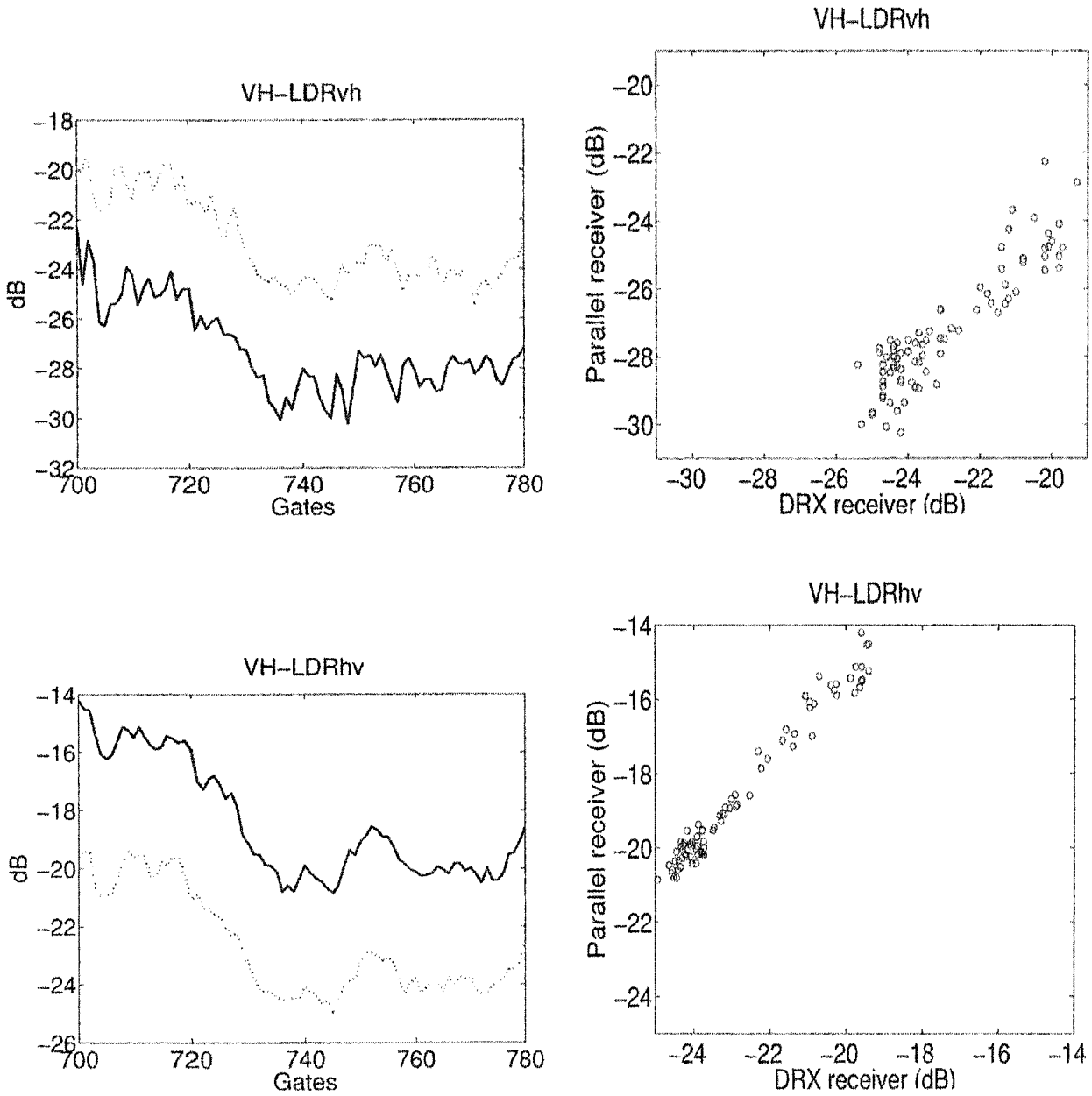


Figure 2.31: LDR_{vh} and LDR_{hv} are compared between the parallel receiver and the DRX receiver. CSU-CHILL radar was operated in alternating mode.

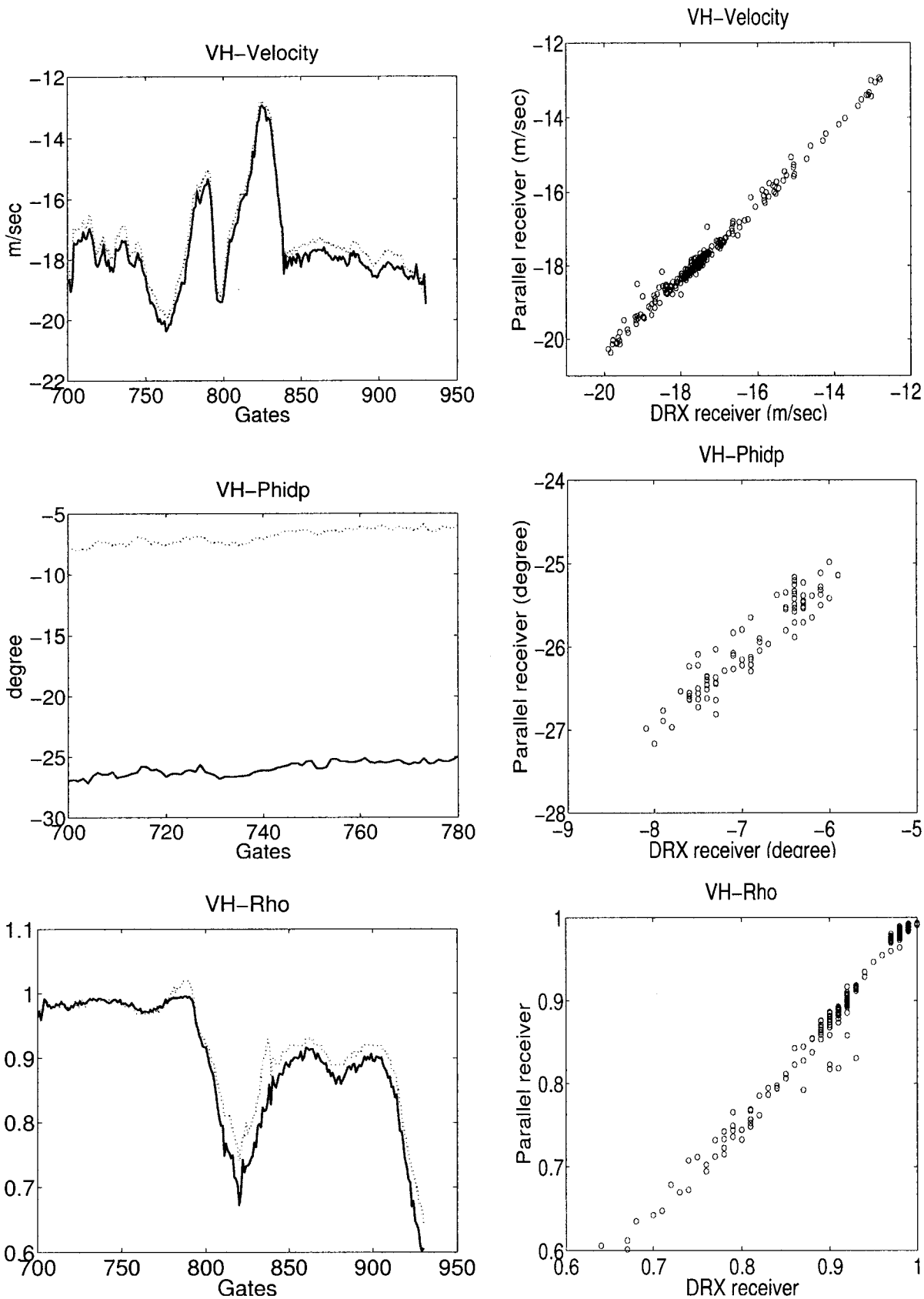


Figure 2.32: Mean velocity, Phi_{dp}, and ρ_{hv} are compared between the parallel receiver and the DRX receiver. CSU-CHILL radar was operated in alternating mode.

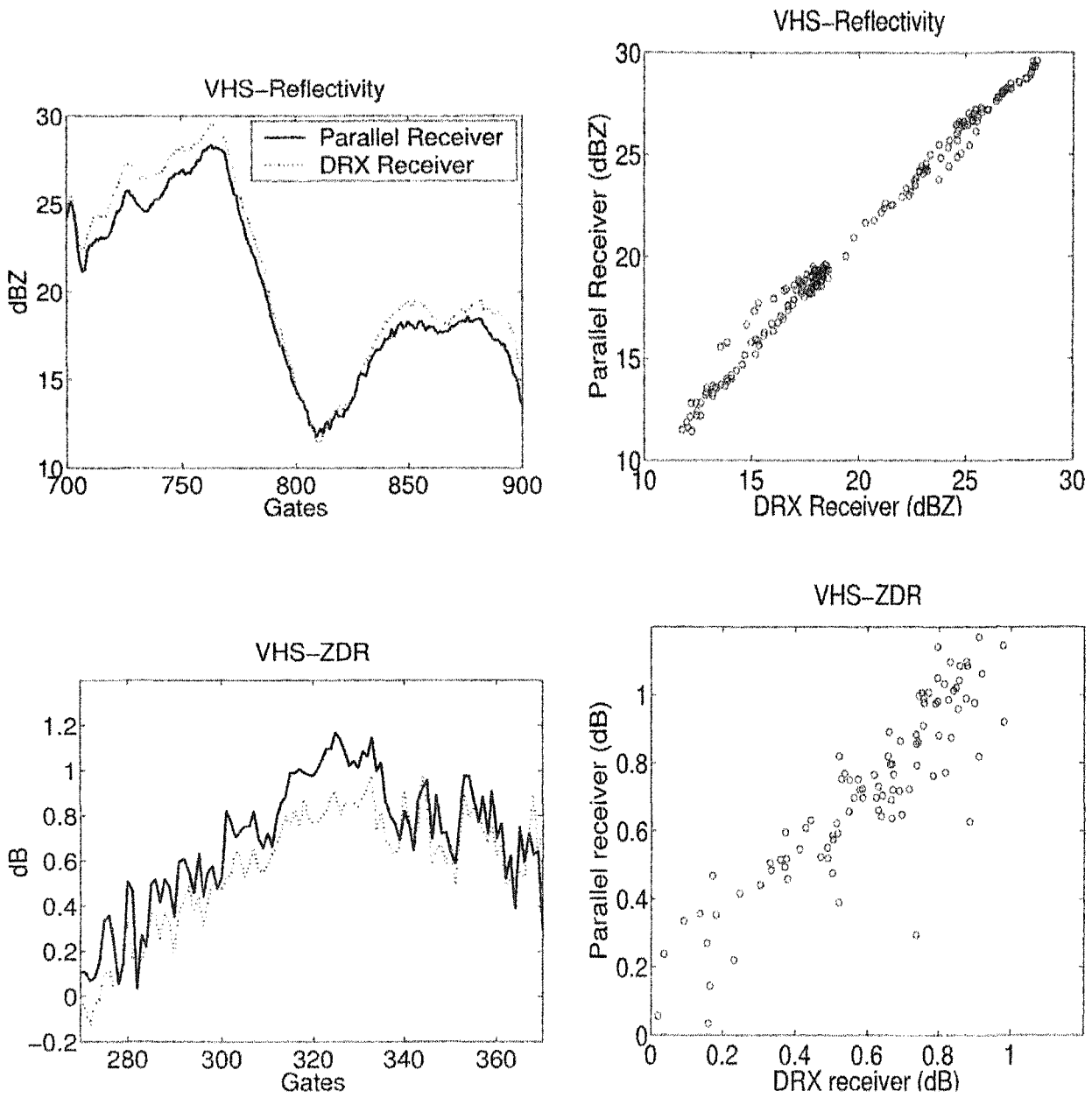


Figure 2.33: Reflectivity and Z_{dr} are compared between the parallel receiver and the DRX receiver. CSU-CHILL radar was operated in hybrid mode.

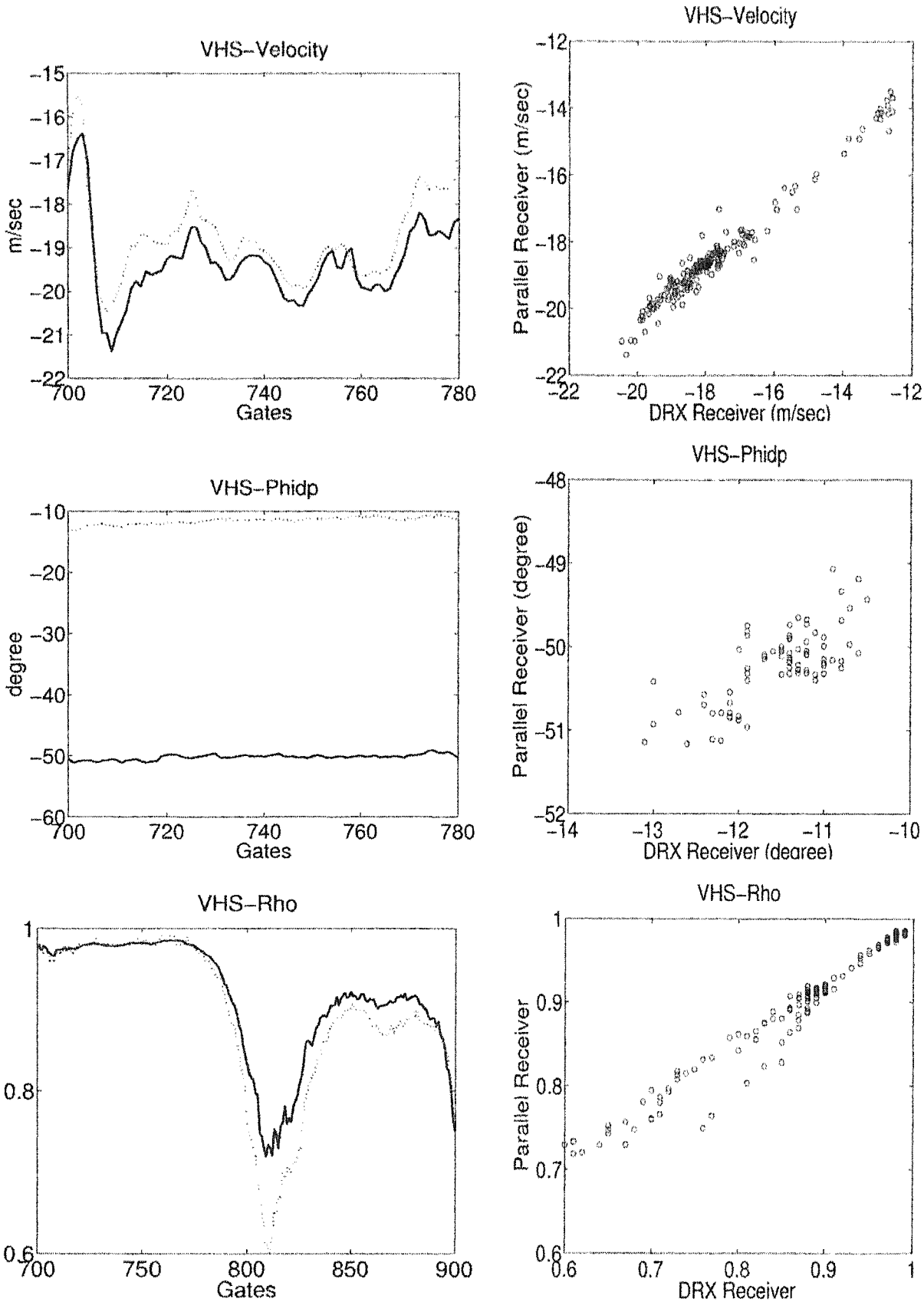


Figure 2.34: Mean velocity, Phi_{dp}, and ρ_{hv} are compared between the parallel receiver and the DRX receiver. CSU-CHILL radar was operated in hybrid mode.

Decimation factor	Sampling rate (MHz)	Transmission rate (Mbps)
40	1	61.6
20	2	121.7
40/3	3	184.5
10	4	245.9
8	5	307.4

Table 2.5: Change of transmission rate with the increase in the sample output rate.

2.5.4 Measurements of transmission rate

The DRS server is designed for the real-time transmission of the DRS to the remote sites. The DRS transmission process, which runs simultaneously with the data acquisition process on the DRS server, is responsible for the transmission. Since the external sampling clock is 40 MHz and the lowest decimation factor is 8, the highest sample output rate possible is 5 MHz currently. The transmission rate over a Gigabit Ethernet was measured increasing the sampling rate up to 5 MHz. During this measurements, the DRS client just received the data and discarded them without estimating radar parameters. Socket buffer size and maximum transmission unit (MTU) were set to be 64 KB and 1500 bytes, respectively. Table 2.5 shows that the transmission rate increases linearly in proportion to the output sampling rate.

2.6 Summary and Conclusion

The digital-IF receiver developed primarily for the high-bandwidth VCHILL is described. The receiver operates in parallel with the DRX processor of the CSU-CHILL radar. The design of the parallel receiver is based on a standard PC platform with PCI-interfaced high-speed ADC/DDC signal processor and a digital I/O, as well as a data conversion circuit for AZ/EL. The parallel receiver

functions as a component of the DRS server, and provides the digitized I and Q components of the received signal of the CSU-CHILL radar with information required for estimation in a specified format. Since the overall data rate that the signal processor generates are over a few hundreds Mbps in conjunction with the operation of the pulsed-Doppler radar, the integration of the all received data into the specified format requires extremely careful timing for synchronized data acquisition and high-speed data manipulation. Optimization of the computer system for the high-speed transfer of data from the FIFO to buffer and the well-organized code are critical for the satisfactory operation of the parallel receiver. In addition, the ADC and DDC processing should be synchronized externally with the pulse transmission. The parameters obtained from the parallel receiver show good coincidence with those obtained from the DRX receiver, which demonstrates that the parallel receiver operates properly for the high-bandwidth VCHILL. The highest sampling rate is 5 MHz currently, and the transmission rate to the remote sites increases linearly in proportion to the sampling rate. The sampling rate can be increased up to 10 MHz with use of the maximum sampling clock of 80 MHz.

Chapter 3

END-SYSTEM ARCHITECTURES FOR THE HIGH-BANDWIDTH VCHILL

3.1 Introduction

The high-bandwidth VCHILL is an approach to transmit the Digitized Radar Signal (DRS) over data network to multiple remote locations where the radar parameters are estimated and delivered to other display nodes in real-time. In a multi-user environment, different users may be interested in different sets of end products. The quality of the end products may depend on the algorithm applied for the processes too. This networked approach to the distribution of real-time DRS provides multiple users with the same quality of the digitized radar signal as at the radar site. Moreover, this approach enables multiple users to independently process the same digitized output signals for various applications [3]. Such an environment demands not only sustained bandwidth requirements of a few hundred Mbps which is much higher than the low-bandwidth VCHILL, but also end systems with high computing capability.

The preliminary implementation of architecture for the DRS data transmission laid foundation of development [21]. The preliminary design was a simulation test system lacking in the details of the radar processes because of an initial trial. Further developments in the end-system architectures, which were designed based

on the operating principle of the pulsed-Doppler radar and the parameter estimation theory, are shown in [9]. The performance was evaluated using a simple test bed via emulation.

In this chapter, we propose end-system architectures for making the notion of high-bandwidth VCHILL possible in conjunction with the transmission of DRS over Transmission Control Protocol (TCP). The architecture is designed based on the client-server model relying on the operation of multiprocesses and multithreads. The key functionalities of the architectures are divided into multiple processes performing the digitized radar signal acquisition and transmission of the DRS on the DRS server, as well as the radar parameter computation and parameter transfer on the DRS client. In addition, generic packet structures and data structures for the information sharing between processes are also designed. The digital-IF receiver described in Chapter 2 is an essential component of the digitized radar signal acquisition. The performance of the proposed architecture implemented on a standard PC/Linux platform is evaluated by using test bed with gigabit link, while operating the CSU-CHILL radar in real-time. Factors that affect the end-to-end TCP throughput are also examined via emulation. The proposed architectures establishe a basic framework for the applications of the high-bandwidth DRS transmission.

3.2 Digitized Radar Signal and Parameters

The CSU-CHILL radar is a dual-polarized radar system such that it can transmit four types of polarization sequences, namely horizontal (H) only, vertical (V) only, alternating between horizontal and vertical, and hybrid mode which contains V and H components simultaneously. In addition, the CHILL radar has two coherent receivers, namely vertical (V) and horizontal (H) receivers [1]. The digitized I and Q components of the received signal are produced on each receiver. It is

well known that the alternating mode and the hybrid mode can provide extensive set of additional parameters in comparison with the H or V only transmit mode [12]. Therefore, we will primarily focus on the radar operation with the alternating mode and the hybrid mode.

The CSU-CHILL radar transmits a pulse at intervals of pulse repetition time (PRT). The receiver digitizes the received signal with a few MHz clock during PRT after every pulse is transmitted. The digitized data set for each transmitted pulse is referred to as a range sample data set. At a particular pointing angle, the pulses are transmitted multiple times to obtain reliable data sets. A collection of the multiple range sample data sets will be referred to as a ray DRS block. Based on this operating principle of pulsed-Doppler radar and synchronized digitizing, we can conceptually visualize the ray DRS block as a two-dimensional array of data as shown in Fig. 3.1. In one dimension, the rows are regularly spaced range samples of the return signal from individually transmitted radar pulses, which are referred to as 'gates'. The position of the data in each row indicates the distance of the object from the radar. When the next pulse is transmitted, the return signals are digitized and stored in the next row of the array. Therefore, the array has a number of rows equal to the number of pulses transmitted and a number of columns equal to the number of samples taken from the return signals.

At remote sites, the DRS client estimates the covariance matrix of the received signal vector, which requires high computing capability. Given these outputs, various radar parameters, such as reflectivity (Z_h), mean velocity (v), normalized spectral width (w), normalized magnitude of autocorrelation with lag one (R_1), differential reflectivity (Z_{dr}), linear depolarization ratio with horizontal transmission (LDR_{vh}), linear depolarization ratio with vertical transmission (LDR_{hv}), differential phase ($\hat{\Psi}_{dp}$) and copolar and crosspolar correlation coefficients (ρ_{hv}, ρ_{cx}), are estimated. Note that the availability of the radar parameters vary depending

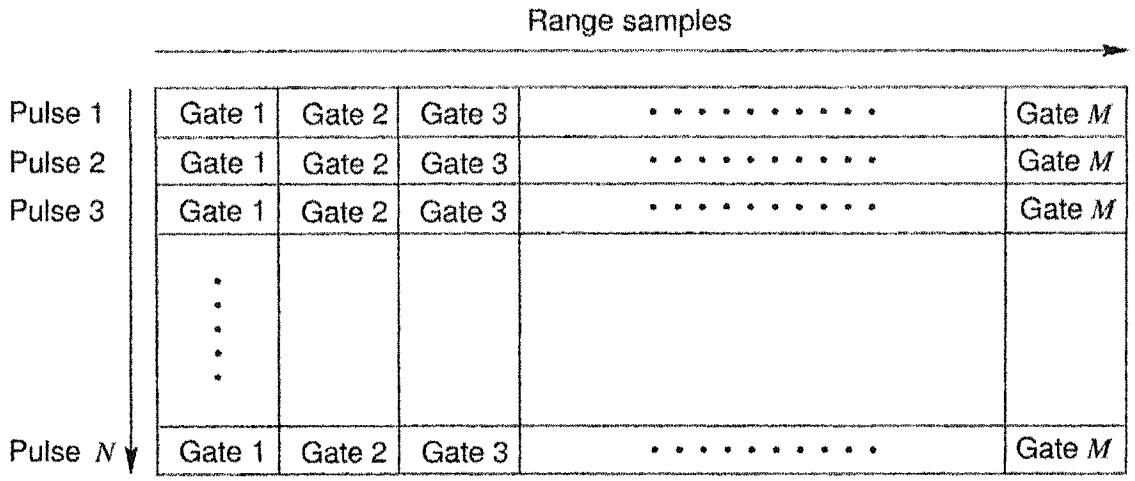


Figure 3.1: A ray DRS block generated when the radar emits N pulses and the receiver digitizes M gates at a pointing angle. Digitized signal attribute varies depending on the radar operating mode. For the alternating mode, the all gates have the copolar sample, i.e. I_{vv} and Q_{vv}, and the cross-polar sample, i.e. I_{hv} and Q_{hv}, for the pulses $(2n-1)$. They have the cross-polar samples, i.e. I_{vh} and Q_{vh}, and the copolar samples, i.e. I_{hh} and Q_{hh}, for the pulses $(2n)$. Here, $n = 1, 2, ..., N/2$. For the hybrid mode, all the gates have I_{vv}, Q_{vv}, I_{hh}, and Q_{hh} for each pulse.

on the radar operating mode. For example, the hybrid mode does not provide the linear depolarizations. Algorithms for the parameter estimation are provided in Appendix C. More detail description about radar parameters can be also found in [12].

The output from one integration cycle is referred to as a ray data set at a particular pointing angle. The collection of all the rays output for a single fixed elevation angle or Azimuthal angle is referred to as a sweep. The collection of a complete set of sweeps is referred to as a volume. The identification numbers of ray, sweep, and volume are delivered to the DRS server through the serial port, and included into the header structure variables. These information are involved in the parameter estimation after being transmitted to the DRS client.

3.3 Design of the end-system architectures

3.3.1 An overview of the architectures

The design is aimed at the DRS transmission through a high-bandwidth data network, such as Next Generation Internet (NGI) or Gigabit LAN. The end-system architectures of the DRS server and client are designed based on the client-server model comprising multiprocesses and multithreads as shown in Fig. 3.2. On the DRS server, the DRS acquisition process reads the DRS and the radar operating conditions, and passes them to the DRS transmission process through shared memory. The DRS transmission process establishes an end-to-end connection to the DRS client, and sends the formatted data using Transmission Control Protocol (TCP) over the high-bandwidth network after reading them from the shared memory. Shared memory and message queue are used for interprocess communications at the end systems, and global variables are used for data exchange between multiple threads [22]. An user payload for transmission is either a ray header or a range sample data set. On the DRS client, the DRS receive process estimates the radar parameters from the received data consisting of the DRS and radar operating conditions, and writes the estimated parameters into the shared memory. The parameter transmission process sends the radar parameters to other nodes for display or further applications over the multicasting protocol after reading them from the shared memory. Primary functions of each process are listed in Table 3.1.

3.3.2 DRS acquisition process

The DRS acquisition process consists of three threads that operate simultaneously for the real-time operations as shown in Fig. 3.2. Since the detail description on the DRS acquisition process and the functions of the threads were provided in Section 2.4, essential functions will be explained from a perspective of the DRS server operation over the data network in this subsection.

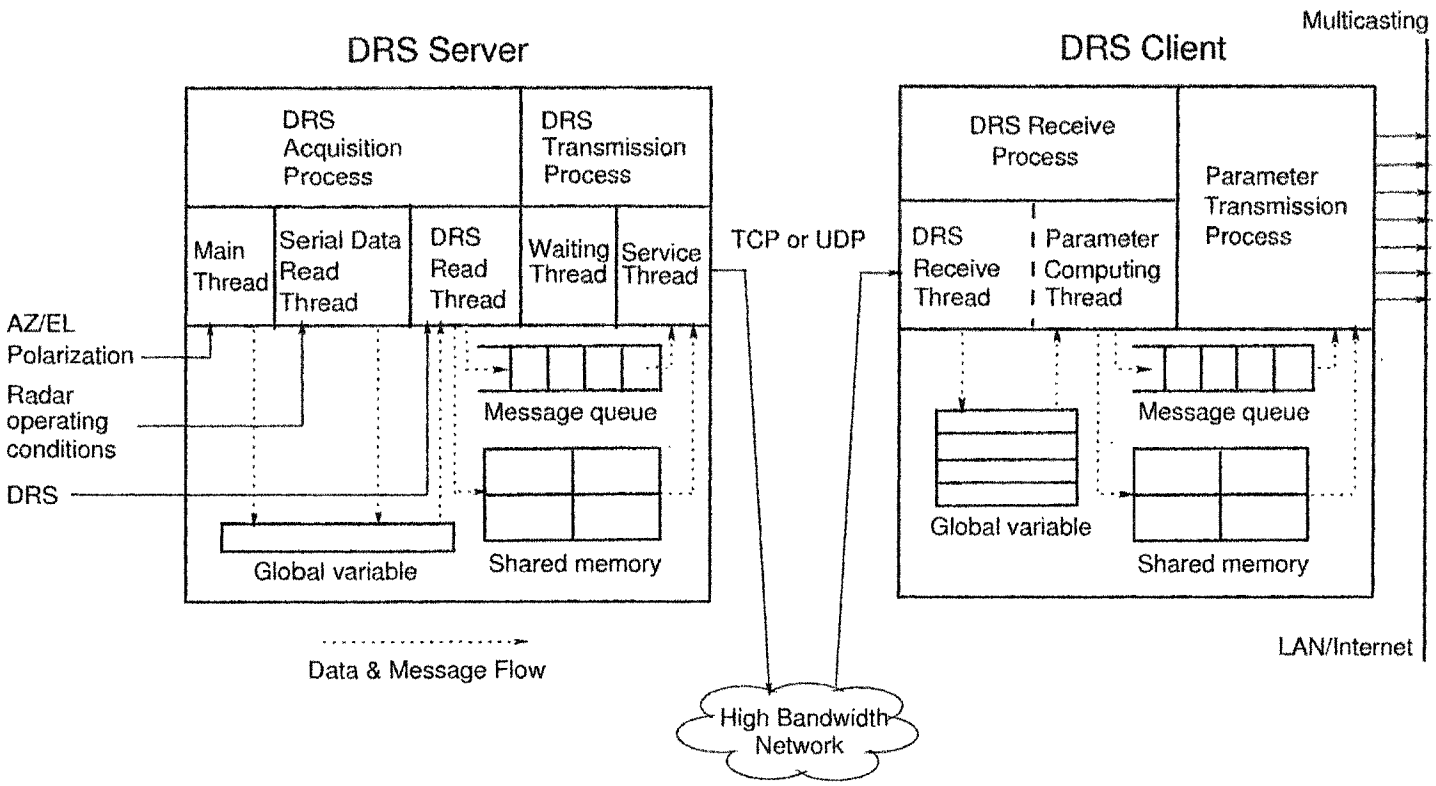


Figure 3.2: Overall end-system architectures for the transmission of the digitized radar signal, the computation of radar parameters, and the delivery of parameters to multiple display nodes and further applications.

Processes	Functions
DRS acquisition	• Initialize the ADC/DDC signal processor and the digital I/O. • Read radar operating conditions. • Read digitized radar signal. • Assemble header information. • Write the formatted data into shared memory.
DRS transmission	• Create listening TCP socket. • Create threads for the service to the accepted client through the connection TCP socket. • Book keeping for all client socket information. • Read the formatted data from the shared memory. • Transmit the formatted data.
DRS receive	• Create TCP socket and establish connection to the DRS server. • Request and get the DRS data. • Estimate the radar parameters. • Calibrate the estimated parameters. • Scale the radar parameters for display. • Make header information for parameter transmission. • Write the radar parameters into shared memory.
Parameter transmission	• Make UDP sockets for the transmission of parameters over multicasting protocol. • Read header information and the radar parameters from the shared memory. • Make user payload which include the display header and the scaled parameters. • Transmit the packets to multicating groups.

Table 3.1: Functions of each process designed for the real-time DRS data transmission and computation.

The DRS read thread assembles two types of headers, namely ray header and range sample header, that are used for radar parameter estimation and display at the remote sites. The ray header is assembled once at a particular pointing angle when the first pulse is transmitted, which specifies the integration cycle in estimating the parameters. However, the range sample header is assembled every time a digitized data set for an individual pulse is read. The structures of the ray header and the range sample header are shown in Table 2.3 and 2.4, respectively. The meaning of the fields can be understood by their names, and will be explained in detail if necessary. While the size of the ray header is constant, the size of a range sample data set depends on the number of gates.

The DRS read thread passes the ray header and the range sample data set to the service thread of the DRS transmission process through a shared memory block. Once the memory is mapped into the address space of the processes that are sharing the memory region, no kernel involvement occurs in passing data between the processes [22]. The shared memory block for data passing consists of four segments. Size of each segment is designed so that each segment can accomodate the largest size of the range sample data set. Data is written into the segment in a circular manner. That is, after a ray header or a range sample data set is written into an available segment, the next data set is written on the next segment and so on. The ray header is followed by the consecutive range sample data sets.

Message queue is utilized for communicating between the DRS acquisition process and the DRS transmission process [22]. After writing a new data on an available segment, the DRS read thread writes the segment number in which the data has been written into the message queue. Another shared memory block is occupied by a structure variable indicating the status of the segment for the data passing. The members of the structure variable indicate information about availability of segment for writing a new data and how many service threads in

the DRS transmission process have read the data. After writing a new data into a segment, the DRS read thread sets the corresponding array element of the member to be one indicating the availability of the segment, which implies that the segment keeps a new data to be read.

3.3.3 DRS transmission process

The DRS transmission process is responsible for sending the ray header and the range sample data sets to the DRS client upon the receipt of request. It consists of a waiting thread and multiple service threads. The number of the multiple service threads depends on a combination of the number of connected DRS clients at a time and the available bandwidth of a physical link which the DRS server is connected to.

The waiting thread creates a TCP socket and waits for requests from the DRS clients. Whenever a client connects to the DRS server, a service thread is spawned from the waiting thread. After passing the receiver's socket information to the service thread, the waiting thread waits for another connection. The service thread creates a new socket and transmits the data to the DRS client. Since thread creation can be 10 - 100 times faster than process creation, the concurrent server model, which is one thread per client [23], is utilized for our development.

The service thread fetches the message that specifies the segment at which the new data is located from the message queue. The service thread checks the element of the structure array member corresponding to the segment number noticed. If the element value is one, the thread reads the data in the segment and sends them to the DRS client. After reading the data stored in the segment, the service thread increments the corresponding array element of the member by one, which points out how many service threads have read the same data in the multiple clients environment. The service thread that has read the data at last sets the

array element of the member to be zero, which notifies that the segment is again available for the DRS acquisition process to write a new data.

As described in the previous subsection, either a ray header or a range sample data set is transferred in a shared memory segment. Therefore, a data transmission unit (a user payload) is either a ray header or a range sample data set. While the size of the ray header is constant, the size of the range sample data set varies depending on the number of gates. Since the ray header keeps the number of gates, the size of the range sample data set can be resolved by reading the ray header.

The service thread checks the type of user payload before sending them by looking at the header identification. It is important to make sure that the first data transmitted to a client should be the ray header after establishing the connection. Then the first, the second and the third range sample data sets are transmitted sequentially until the last one is sent. After completion of transmission of a ray DRS block, the next ray header is transmitted followed by the range sample data sets and so on. Without the ray header information, the following range sample data sets cannot be used for estimating radar parameters at the remote sites, which just results in the waste of the network bandwidth.

3.3.4 DRS receive process

3.3.4.1 Data receive thread

The DRS receive thread requests the DRS data transmission to the server after establishing TCP connection. The first data that the DRS receive thread should get is a ray header followed by the sequentially delivered range sample data sets. The DRS receive thread accumulates the received data in a segment of buffer until a ray DRS block is assembled. This thread also prepares an array variable that becomes a part of the ray DRS block. When the thread receives a

range sample data set, it sets the corresponding element to be one after reading the field of the data number in the range sample header, which indicates the range sample data set has been received. If any range sample data sets are missing, the corresponding array elements are set to be zero. For the transmission over TCP, because of the guaranteed data transmission the check is not necessary. However, for the transmission over unreliable transport layer protocol, such as UDP, the information about missing range sample data sets should be reflected on the estimation of the parameters. Eventually, a ray DRS block is composed of a ray header, an array variable, and range sample data sets as illustrated in Fig. 3.3. A ray DRS block is handled as a data unit for the estimation of the parameters.

The buffer that keeps the ray DRS block consists of four segments. The ray DRS block is accumulated in the segment and read in the circular manner. A global array variable is used for interchanging information between the DRS receive thread and the parameter computing thread. Once a ray DRS block is stacked in a segment, the corresponding array element is set to one, which notifies the parameter computing thread the availability of a new ray DRS block. After all radar parameters are computed, the parameter computing thread resets the element, releasing the segment free. Then, the DRS receive thread can refill the segment with a new DRS ray block.

3.3.4.2 Parameter computing thread

The parameter computing thread estimates the radar parameters with the ray DRS block read from the buffer. The estimation process is performed gate by gate. Given a ray DRS block, a covariance matrix for a gate is estimated along the sample time. Then radar parameters are computed with the estimators of the covariance matrix according to the equations described in Appendix C. Iterating the above procedure over the all gates produces the radar parameters for a ray. Next, the

Ray header				
Range sample data set check array				
Range sample 1 header	Ivv Qvv Ihv Qhv	Ivv Qvv Ihv Qhv	• • • • •	Ivv Qvv Ihv Qhv
Range sample 2 header	Ivh Qvh Ihh Qhh	Ivh Qvh Ihh Qhh		Ivh Qvh Ihh Qhh
• • •				
	Ivv Qvv Ihv Qhv	Ivv Qvv Ihv Qhv		Ivv Qvv Ihv Qhv
Range sample N header	Ivh Qvh Ihh Qhh	Ivh Qvh Ihh Qhh	• • • • •	Ivh Qvh Ihh Qhh
	gate 1	gate 2		gate M

(a) Alternating mode

Ray header				
Range sample data set check array				
Range sample 1 header	Ivv Qvv Ihh Qhh	Ivv Qvv Ihh Qhh	• • • • •	Ivv Qvv Ihh Qhh
Range sample 2 header	Ivv Qvv Ihh Qhh	Ivv Qvv Ihh Qhh		Ivv Qvv Ihh Qhh
• • •				
	Ivv Qvv Ihh Qhh	Ivv Qvv Ihh Qhh		Ivv Qvv Ihh Qhh
Range sample N header	Ivv Qvv Ihh Qhh	Ivv Qvv Ihh Qhh	• • • • •	Ivv Qvv Ihh Qhh
	gate 1	gate 2		gate M

(b) Hybrid mode

Figure 3.3: A ray DRS block assembled in a segment of buffer on the DRS client. Although the block format is identical, the data attributes are different depending on the radar operating mode. (a) Alternating mode and (b) Hybrid mode.

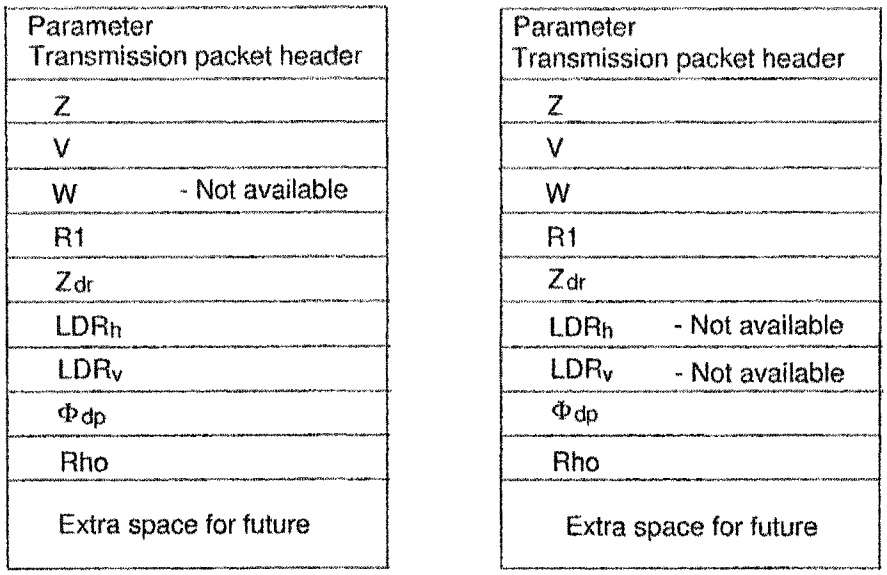


Figure 3.4: Parameter data block structure. Available radar parameters change depending on the radar operating mode. The size of parameter transmission packet header is 148 bytes. The size of one radar parameter data is determined by the number of gates, in which one gate data occupies 1 byte.

parameters are calibrated and scaled for the display. Appendix D describes the equations for the calibration in detail. This period of parameter estimation for a ray is referred to as an integration cycle.

The parameter computing thread also constructs a header based on the ray header information for parameter display. Structure of the parameter transmission header is described in [24]. The structure of the parameter data block consisting of the header and the parameters is shown in Fig. 3.4. The data block is passed to the parameter transmission process through a shared memory segment. The DRS receive process and the parameter transmission process exchange information each other in the same way as those on the DRS server.

3.3.5 Parameter transmission process

The parameter transmission process reads the parameter block from a segment of the shared memory and sends them to the group of end users over multicasting

protocol at a time. The process creates multiple UDP sockets, each of which is associated with a radar parameter. A user payload for the transmission consists of a display header and a radar parameter data set. The process releases the segment after sending all parameters for one ray.

The use of multicasting protocol provides high scalability with respect to the number of end users without any additional resource demands, such as bandwidth and workload for transmission. Each IP multicast group has an identifier called a multicast group ID that is composed of a Class D IP address and a port number [23]. Whenever a packet is sent out from a sender, a multicast group ID included in the the packet specifies the destination group.

3.4 Implementation and performance evaluation

3.4.1 End-to-end TCP throughput

The proposed end-system architectures have been implemented on the standard PC/Linux platform using C language, standard BSD socket interface, System V Interprocess Communication (IPC), and Posix threads. As described in Chapter 2, the DRS server was implemented on a PC containing Xeon dual processors (2.8 GHz) with Red Hat 8.0 Linux operating system. The ADC/DDC signal processor and the digital I/O card are interfaced with a PCI-X (64-bit/66MHz) and a PCI (32-bit/33 MHz) slots, respectively. The APIs, which are provided by the manufacturers to access the ADC/DDC signal processor and the digital I/O card [18, 19], are utilized in implementing the DRS acquisition process. The DRS client program was implemented on a PC containing Xeon dual processors (2.2 GHz) with Red Hat 8.0 Linux operating system.

Figure 3.5 shows the test bed established for the performance evaluation. Before performing the real-time TCP throughput measurement between the DRS server and client, available link capacity of the test bed was determined by the

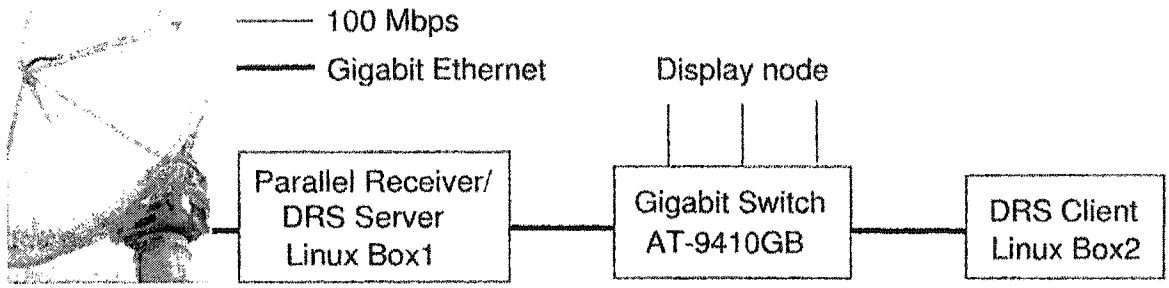


Figure 3.5: Test bed for evaluating performance of the end systems and throughput.

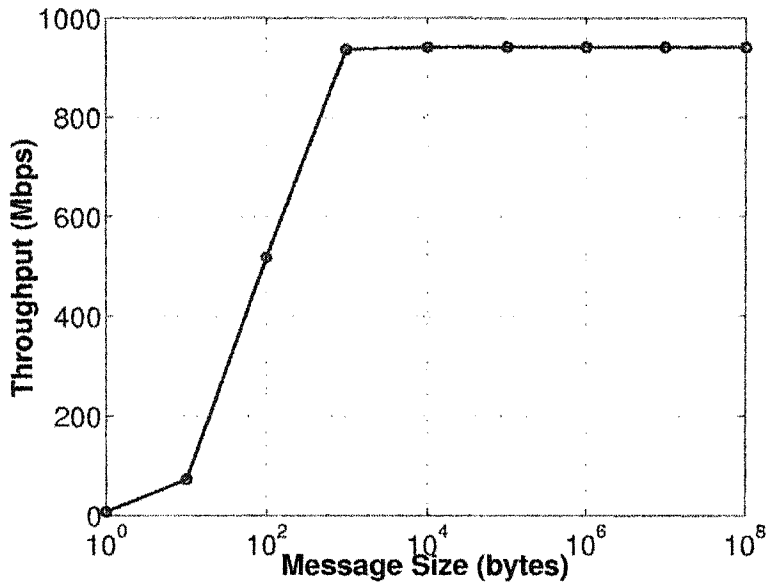


Figure 3.6: Link capacity between the DRS server and client.

Netperf [25]. The most common use of Netperf, which is designed based on the basic client-server model, is measuring bulk data transfer performance, or in other words 'stream' or 'unidirectional stream' performance. Essentially, the test measures how fast one system can send data to another and/or how fast that other system can receive it [25]. In the measurements, sending/receiving socket buffer size and MTU were set to be 64 KB and 1500 bytes, respectively. Figure 3.6 shows the throughput varying the message size, which is equivalent to bulk data size to be transferred.

The end-to-end TCP throughput between the DRS server and client was evaluated using the simple test bed, while operating the CSU-CHILL radar. The sampling rate at the DRS server was varied from 1 MHz to 5 MHz. As described in Section 3.2, the DRS server at 1 MHz and 5 MHz sampling rates approximately generates a row of range sample data set per a pulse transmission whose size are 8,000 bytes and 40,000 bytes, respectively. The server generates intermediate sizes of range sample data set when the sampling rate is between the two sampling frequencies. Since the DRS transmission process write the range sample data set after reading them from the shared memory, the range sample data set set can be referred to be a user payload. When the application program calls 'write', the kernel copies the user payload from the application buffer into the TCP socket send buffer [23]. Here, since the user payload is generated per pulse transmission, the user payload can be equivalent to the message presented in the Netperf measurement. Accoridng to the Netperf results, the maximum throughput achievable with these message sizes is approximately 940 Mbps as shown in Fig. 3.6. In this test bed configuration, the DRS transfer consequently utilizes small portion of the available link capacity in the real-time operation. Table 3.2 shows that the end-to-end TCP throughput, which is measured by using a time-measuring function in the service thread on the DRS server as shown in Fig. 3.2. The throughputs are identical with the data rate generated in the parallel receiver, linearly increase in proportional to the sampling rate within the maximum link capacity achievable. In addition, the throughputs are definitely identical with the results obtained with no parameter estimation at the DRS client. Because of the high computing capability of the DRS client, the parameter estimation does not affect the TCP throughput at all. However, when the computing capability is insufficient for real-time applications, the TCP throughput is affected by the computational workload at the DRS client as described in the following section.

Decimation factor	Sampling rate (MHz)	Transmission rate (Mbps)
40	1	61.6
20	2	122.9
40/3	3	183.4
10	4	242.2
8	5	294.3

Table 3.2: Change of transmission rate with the increase in the sample output rate. The transmission rate is identical to the data rate generated by the parallel receiver.

3.4.2 Display comparison with the DRX system

To evaluate the functionality of the end-system architectures of the high-bandwidth VCHILL, we compared the displays of the radar parameters estimated by the high-bandwidth VCHILL and the DRX system, while operating the CSU-CHILL radar. The displaying pictures were captured simultaneously from the display windows running on Sun workstations. Comparison of the displays show good coincidence between the two displays as shown from Fig. 3.7 to Fig. 3.14.

3.5 Performance evaluation via emulation

3.5.1 Factors affecting the end-to-end TCP throughput

The performance of the proposed end-system architectures were also evaluated on Sun/Solaris platform (Sun Blade 1000/Solaris 8) via emulation to study effect of the end-system computing capability on the end-to-end TCP throughput. The Sun Blade 1000 has dual 750 MHz UltraSPARC-III processors running with Solaris 8. The computing capability of the Sun Blade 1000 is lower than that of the PC with the dual Xeon processors [26]. Gigabit Ethernet Network Interface Card SK-9821 from SysKonnect GmbH is plugged into a 64-bit/66 MHz PCI slot. A simple test bed was established with the similar configuration as the real-time operation

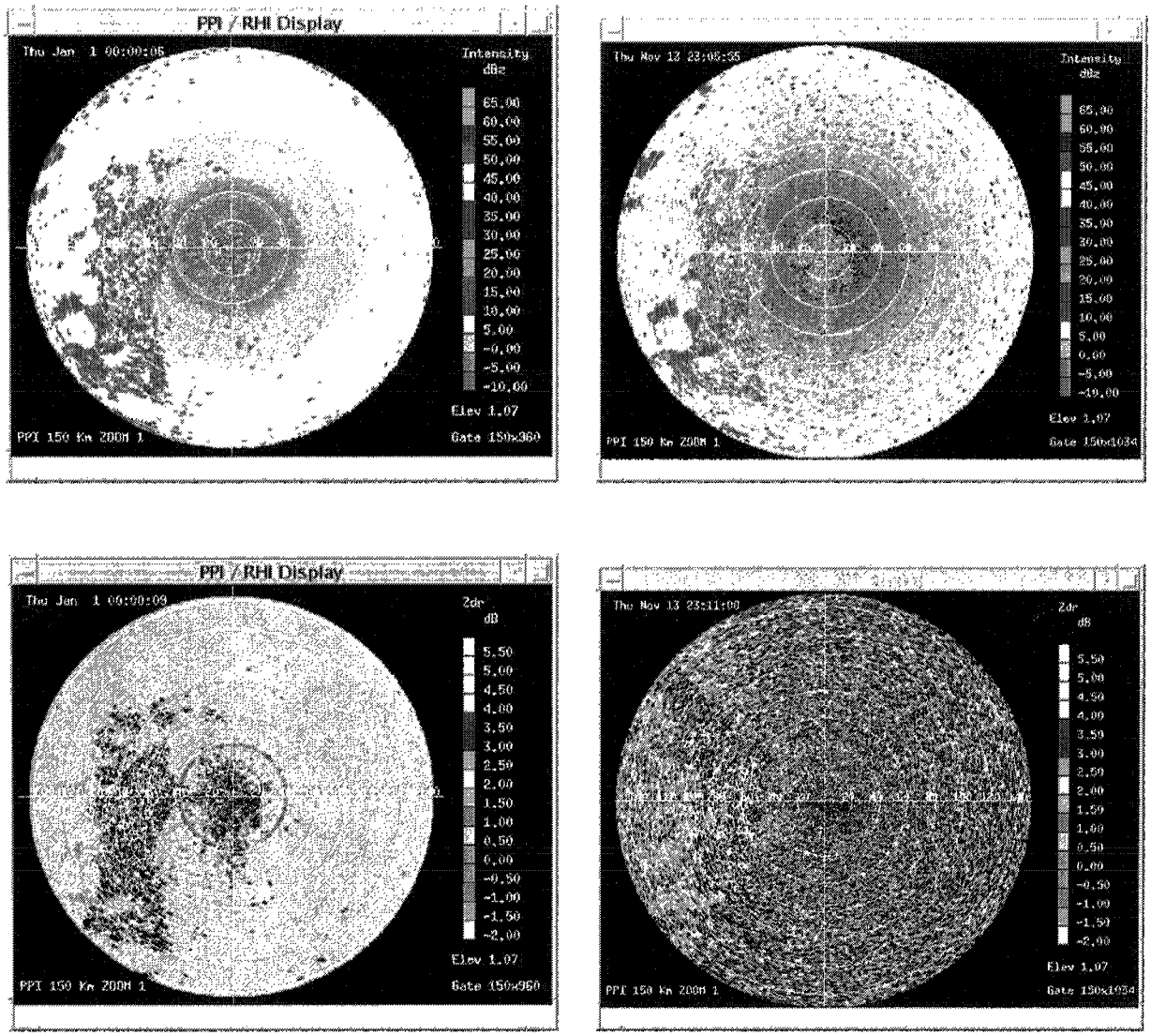


Figure 3.7: PPI-scan displays of Reflectivity and Z_{dr}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

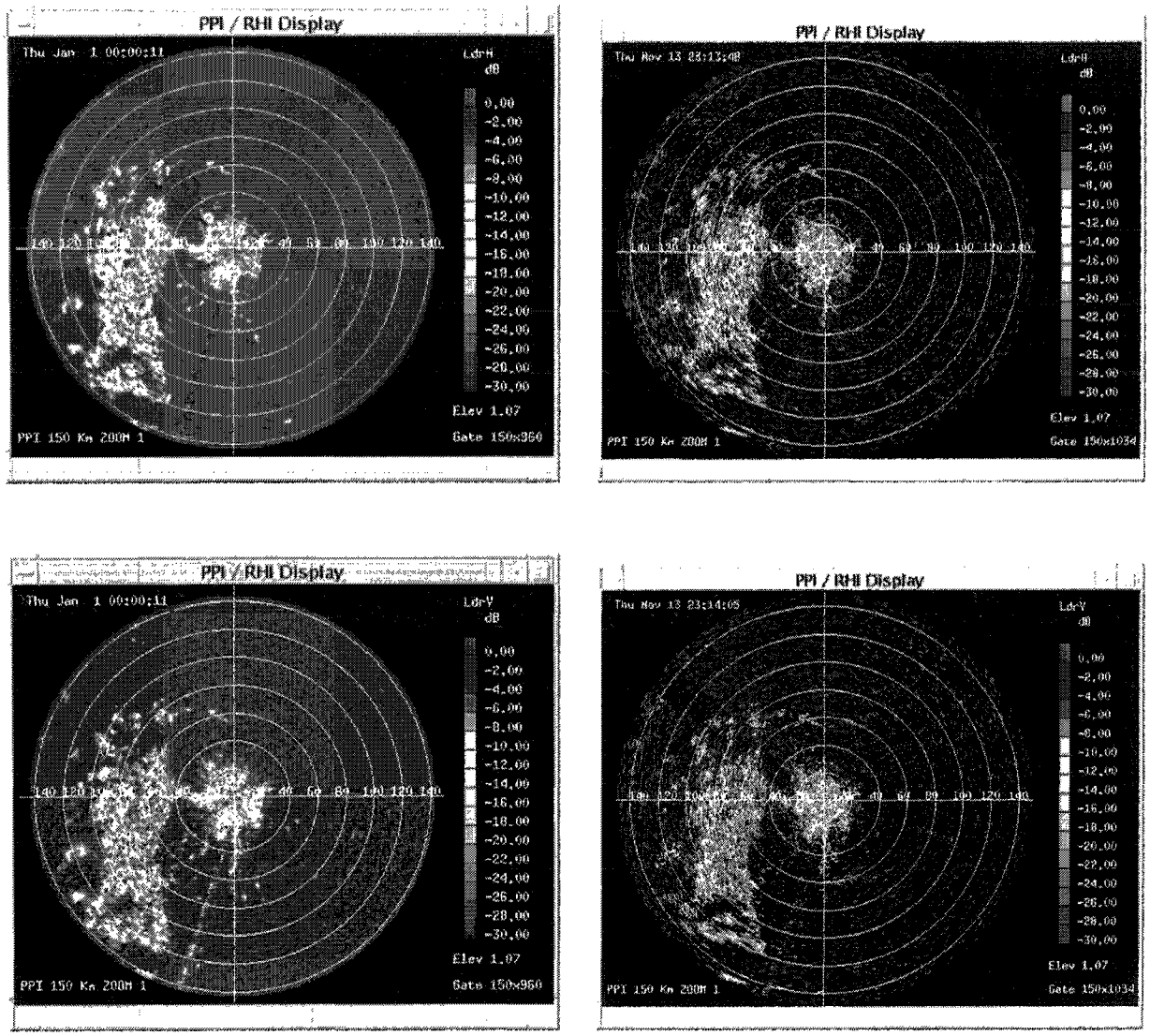


Figure 3.8: PPI-scan displays of LDR_{vh} and LDR_{hv}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

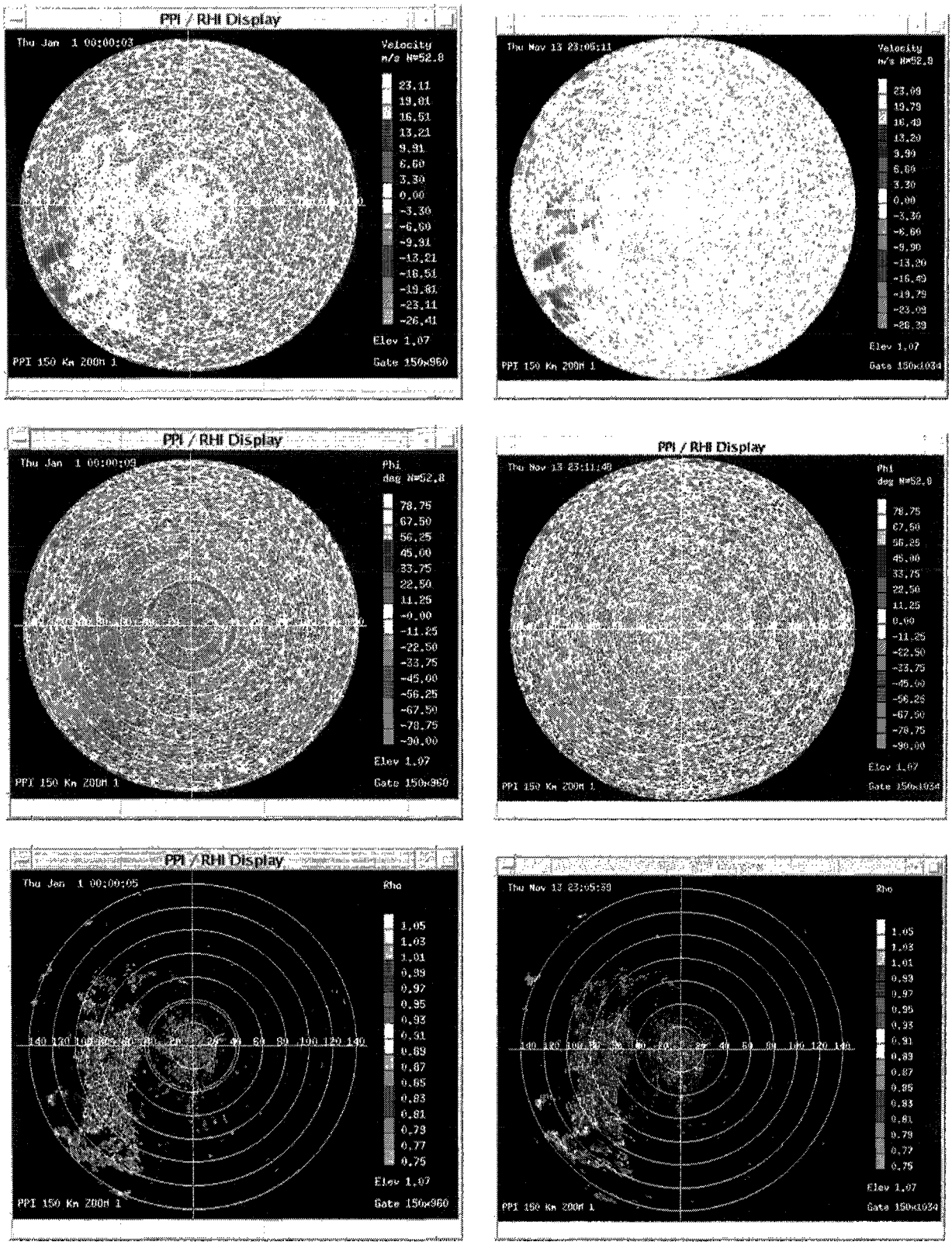


Figure 3.9: PPI-scan displays of mean velocity, Phi_{dp}, and ρ_{hv}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

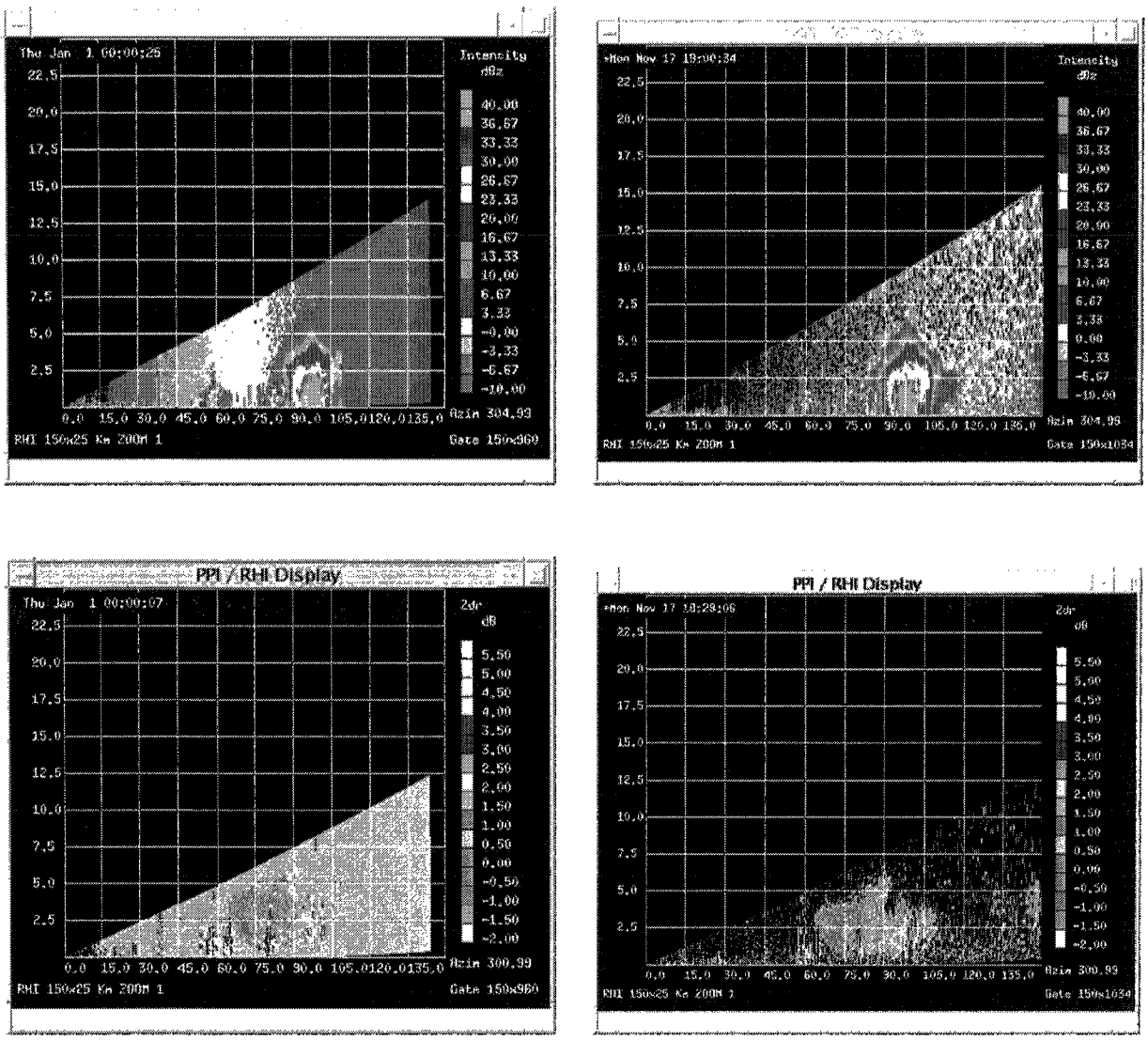


Figure 3.10: RHI-scan displays of Reflectivity and Z_{dr}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

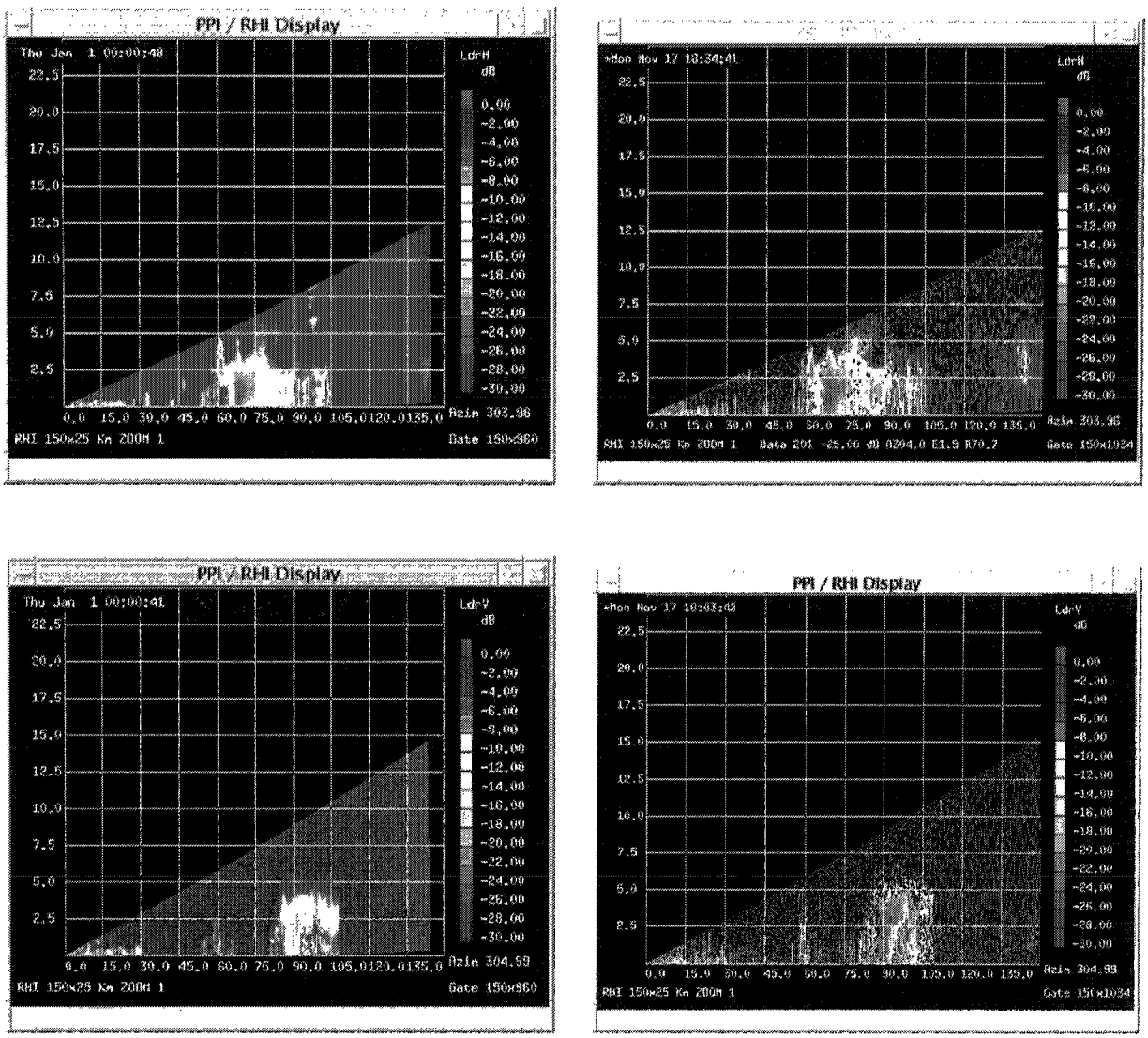


Figure 3.11: RHI-scan displays of LDR_{vh} and LDR_{hv}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

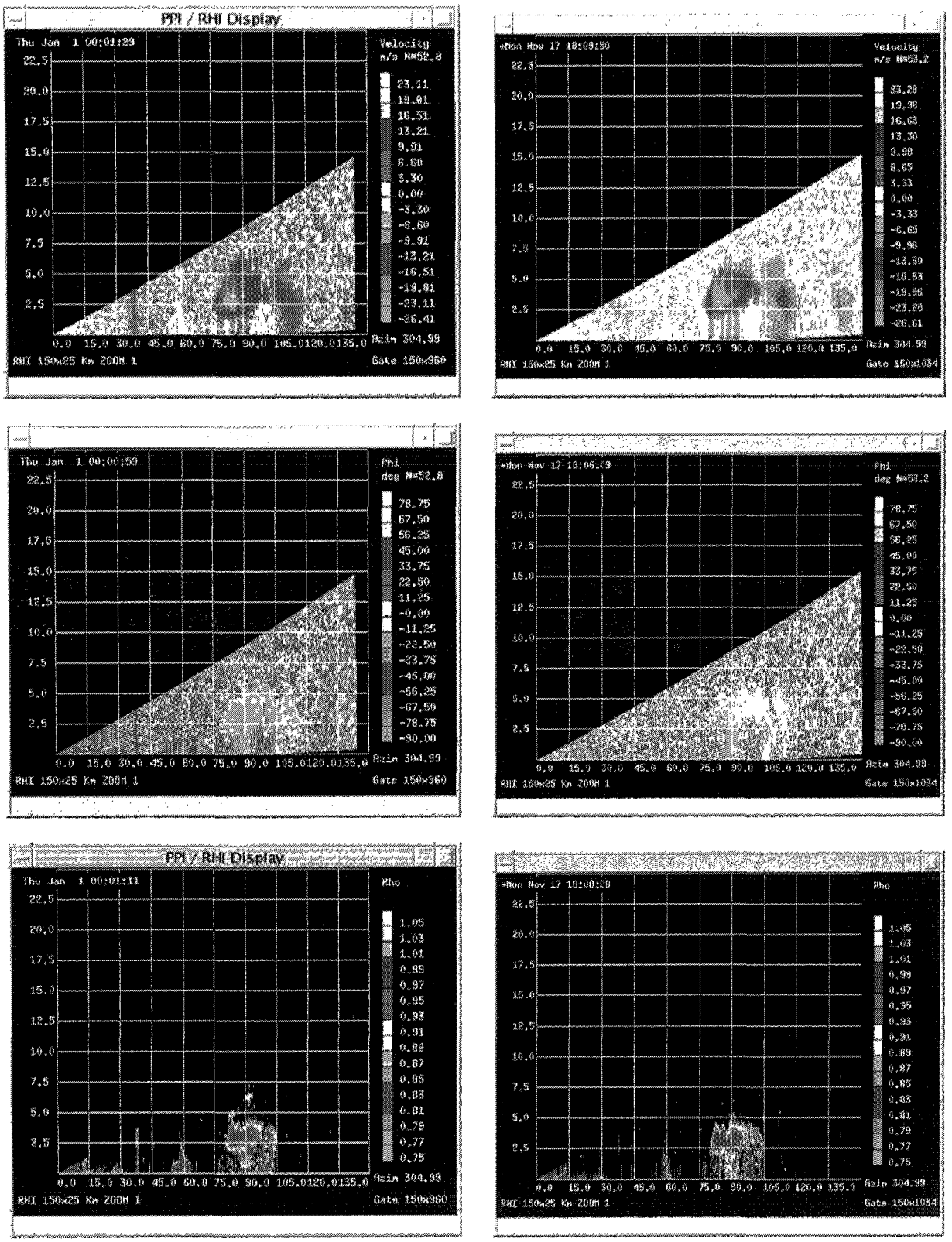


Figure 3.12: RHI-scan displays of mean velocity, Phi_{dp}, and ρ_{hv}. CSU-CHILL radar was operated in alternating mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

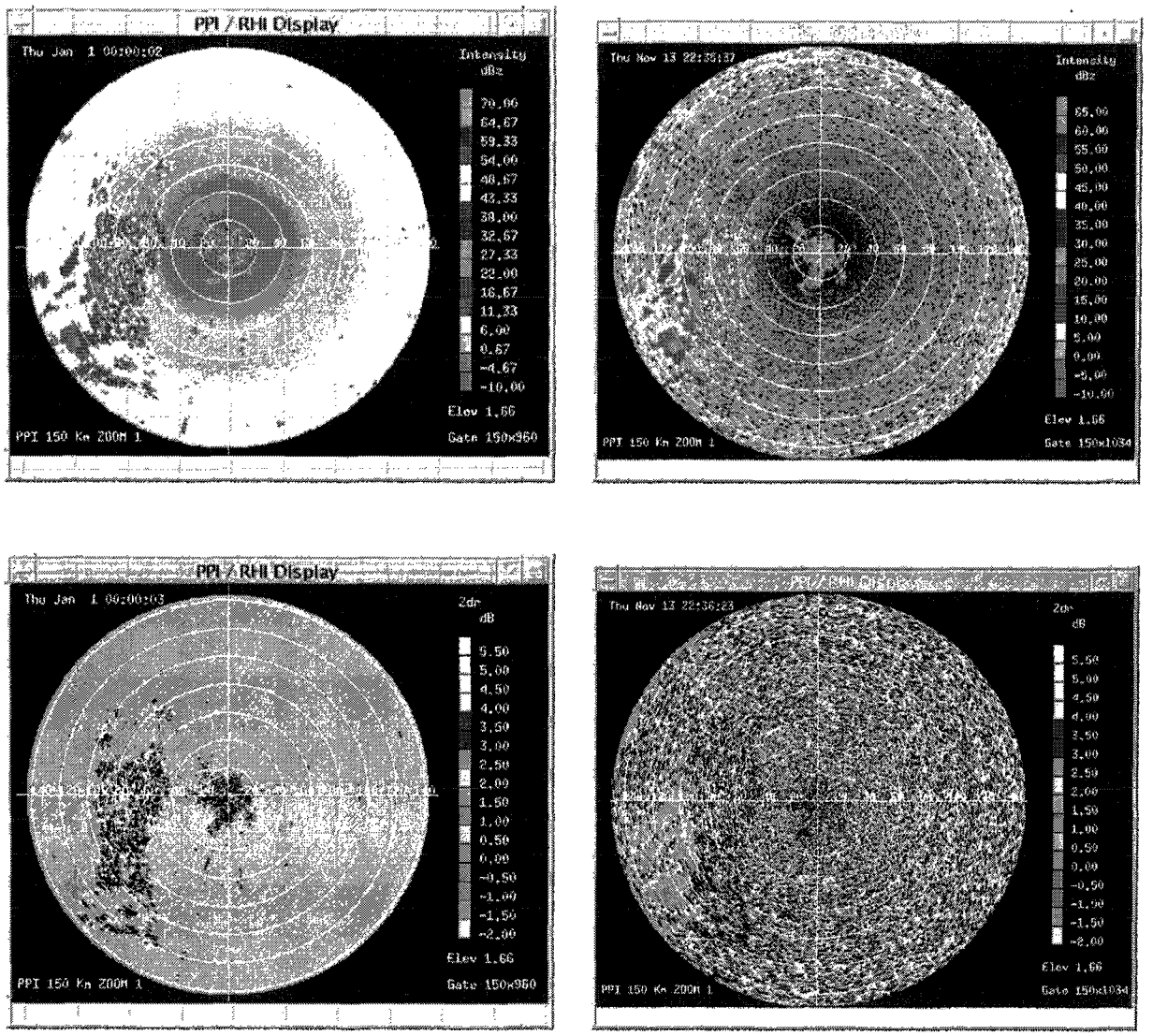


Figure 3.13: PPI-scan displays of Reflectivity and Z_{dr}. CSU-CHILL radar was operated in hybrid mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

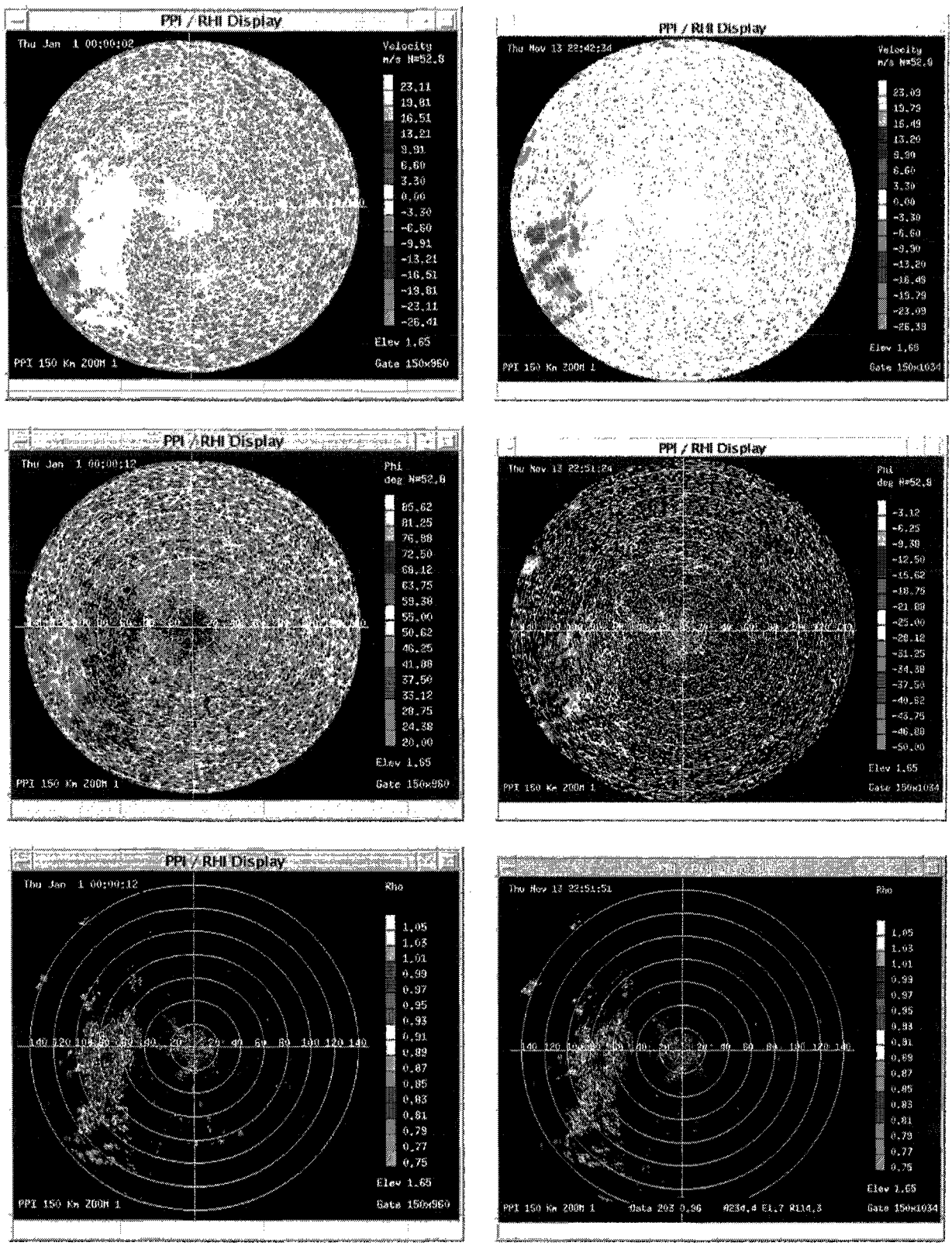


Figure 3.14: PPI-scan displays of mean velocity, Phi_{dp}, and ρ_{hv}. CSU-CHILL radar was operated in hybrid mode. (Left) High-bandwidth VCHILL and (Right) DRX system.

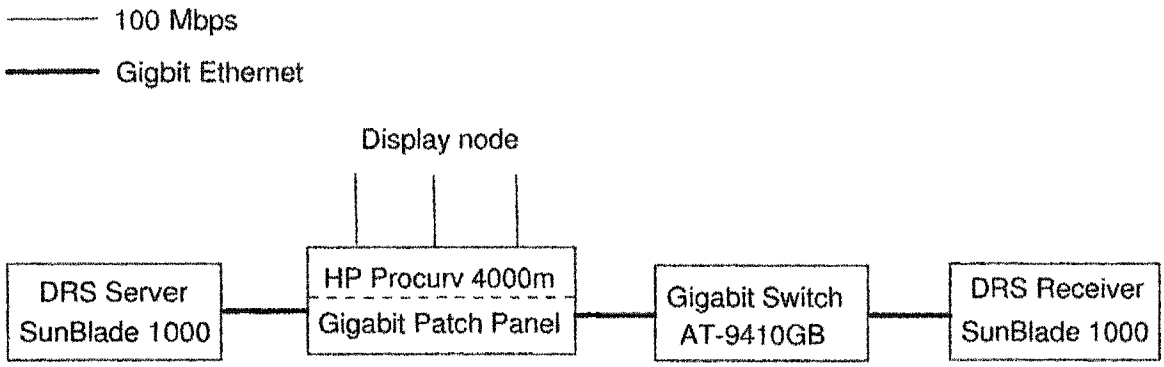


Figure 3.15: Test bed for emulation and evaluation of the performance.

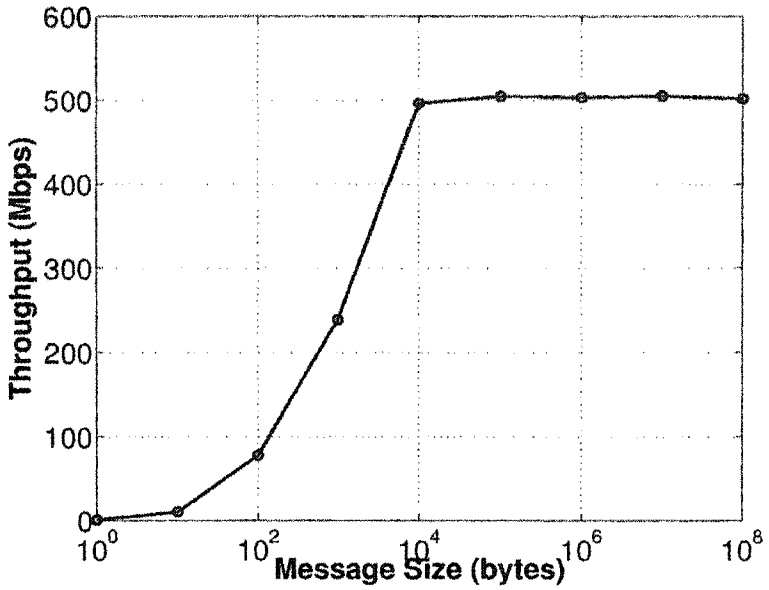


Figure 3.16: Link capacity between the DRS server and client.

as shown in Fig. 3.15. The link capacity of the test bed over TCP, which was obtained by the Netperf, is shown in Fig. 3.16.

For this emulation, a sweep of the real DRS data was prepared as a file. The radar operating conditions were assumed as follows; a) the PRT was 1 msec, b) the number of pulses transmitted at a specific angle was 128, c) the angular resolution of a ray was 0.75 degrees, and the sampling rate was 1 MHz. The DRS data was stored in the server RAM after being loaded. The DRS acquisition process emulated the radar operation by writing the ray header into the shared memory

followed by the 128 range sample data sets sequentially. The end-to-end TCP throughput was 462 Mbps when the parameters were not estimated at the DRS client. This measuring condition is equivalent to the Netperf measurement with the message size of 8000 bytes because the DRS server transmits the data as fast as it can. The throughput measured shows the coincidence in comparison with the available link capacity of the emulation test bed as shown in Fig. 3.16, in which the throughput is approximately 470 Mbps with the message size of 8000 bytes. However, the throughput of 100.9 Mbps was obtained when the parameters were estimated at the DRS client, which shows that the estimation of the parameters reduces considerably the end-to-end TCP throughput even though the throughput is still higher than the assumed data rate.

As explained in Section 3.3.4, the DRS receive thread on the client writes the received data into a segment of the buffer. When there is no available segment, the thread waits for a released segment. The TCP socket buffer on the client is filled up with the delivered data during this time. The segment can be only released after the completion of the parameters of a ray by the parameter computing thread. Consequently, the DRS client cannot receive any data from the DRS server due to no available socket buffer space, which results in the degradation of throughput. During the evaluation of the performance, it was observed that the DRS receive process was utilizing more than 90% of the total available computing resources of the DRS client. Therefore, the computing capability determines the segment release speed and eventually affects the end-to-end TCP throughput.

To study the effect of optimization in compiling on the performance, the DRS receive process was compiled with different optimization levels in the Sun/Solaris platform. Table 3.3 shows the dependency of throughput on the compiling optimization level, which obviously indicates that the computing capability is critical

Optimization level	O0	O1	O2	O3
Throughput (Mbps)	36.5	95.1	99.5	100.9

Table 3.3: Throughputs running the DRS receive processes compiled with different optimization levels of gcc compiler.

to obtain the required TCP throughput too. Compiling with higher optimization level than -O3 did not improve the throughput any more.

Especially in implementing the signal processing thread, we should be very careful in coding. For example, when the parameter computing thread directly handled the DRS data components, such as I_{vv}, Q_{vv}, I_{hh}, and Q_{hh}, the throughput was 100.9 Mbps. However, the use of redundant variables, which makes the code look easy to read, reduced the throughput to 46.4 Mbps. To sum up, preserving high computing capability of the DRS client, developing high speed algorithm for computing, and programming skills, are critical for the successful real-time operation.

3.5.2 Function profile of the data receive process

As observed in the previous subsection, the parameter computing thread is critical for improving the throughput. To study how much time individual function in the DRS receive process occupies, we profiled the process using 'GNU gprof' [27]. The result shows that most time the DRS receive process spent was consumed by two functions. The function calculating the radar parameters and the function finding the available segment into which the thread was going to write the received data, occupy 49% and 47.5% of the total time, respectively. Using time measuring function included in gcc compiler, we also analyzed the percentage of time spent by different parts within the parameter calculation function. As shown in Fig. 3.17, approximately 80% of time was spent for calcuating the covariance estimators. The

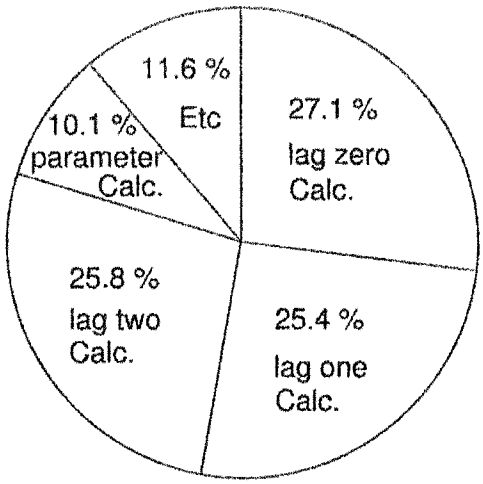


Figure 3.17: Profiling graph.

code for computing the parameters from those covariance estimators consumed around 10%.

A library function of 'mutex' was employed to regulate synchronously the threads access to the global variable segment [22]. At a time, only a thread, which is either the DRS receive thread or the parameter computing thread, has privilege to access to the buffer in which a ray DRS block is stored. For instance, when the parameter computing thread is computing the radar parameters with a ray DRS block that is stored in a buffer, the DRS receive thread cannot write the next ray DRS block into it. Once the global variable is unlocked, the DRS receive thread can write next data into that variable. The result of 'gprof' showed that 92.7% of the running time was occupied by the function calculating radar parameters, which is much longer compared to the case with no use of mutex. This indicates that most time the global variable was locked by the function calculating the radar parameters as we expected.

However, this longer occupation did not contribute to the improvement of throughput, and even reduced the throughput as shown in Table 3.4. Percentage of the CPU occupation by the DRS receive process was also dropped by 15 - 20%. When the covariance estimators of lag 0, lag 1, and lag 2 were calculated, the

average throughputs was reduced from 85.3 Mbps to 79.1 Mbps. Similarly, when the covarience estimators of lag 0 and lag 1 are calculated, the average throughput dropped from 110.7 Mbps to 97.9 Mbps.

Measurements Trials	lag 0, lag 1, lag 2 calculation (no mutex)	lag 0, lag 1, lag 2 calculation (mutex)	lag 0, lag 1 calculation (no mutex)	lag 0, lag 1 calculation (mutex)
1	85.4/70 - 75%	78.1/48 - 51%	117.3/ 61 - 73%	97.7/ 46 - 51%
2	86.1/70 - 76%	78.8/48 - 51%	109.3/ 65 - 73%	96.8/ 47 - 50%
3	84.3/70 - 75%	80.4/49 - 51%	108.9/ 62 - 70%	98.2/ 46 - 51%
4	84.5/70 - 75%	79.6/48 - 51%	109.4/ 62 - 70%	99.1/ 47 - 51%
5	86.1/71 - 76%	78.5/48 - 51%	108.4/ 62 - 70%	97.9/ 46 - 51%
Average Throughput	85.3 Mbps	79.1 Mbps	110.7 Mbps	97.9 Mbps

Table 3.4: Effect of the 'mutex' on the end-to-end TCP throughputs (Mbps) and CPU occupation of the DRS receive process (%).

3.6 Summary and Conclusion

End-system architectures for making the notion of the high-bandwidth VCHILL possible have been described. Design of the architectures is based on the client/server model, relying on the simultaneous operation of multiprocesses and multithreads to meet the real-time requirements. The design includes the data and packet structures too. The architectures include the functions of the DRS acquisition and the transmission of the real-time DRS over TCP to remote sites on the DRS server, as well as the receive/estimation of the radar parameters and the delivery of the parameters to other display nodes on the DRS client. Each function of the operations are divided into individual process. The architecture was successfully implemented on a standard PC/Linux platforms and evaluated in real-time using a test bed, while operating the CSU-CHILL radar. The implemented application and network protocols support the real-time operation of the CSU-CHILL radar

with the data rate of 294.3 Mbps, which is equivalent to the sampling rate of 5 MHz. Comparison of the displays between the high-bandwidth VCHILL system and the DRX system shows coincidence. The performance evaluation via emulation on the Sun/Solaris platform shows that the estimation of the parameters can reduce the end-to-end TCP throughput, which might cause failure in real-time operation. Therefore, preserving high computing capability of the DRS client, developing high speed algorithm for computing, and programming skills, are critical for the successful real-time operation in future applications that require higher computing capability. The end-system architectures developed can be used for the various applications of the high-bandwidth VCHILL.

Chapter 4

RADAR WAVEFORM DESIGN FOR CONGESTION CONTROL IN THE HIGH-BANDWIDTH VCHILL

4.1 Introduction

End-system architectures for the transmission of the high-bandwidth DRS over TCP and the estimation of the radar parameters at remote site were developed in Chapter 3. The server transmits the real-time DRS in a specified format to the client using TCP over high-bandwidth data network. However, the end-to-end TCP throughput is generally determined by segment size, round trip time (RTT) and packet loss rate [28]. Therefore, the end-to-end TCP throughput varies depending on the physical distance between end nodes and network traffic load, which might lead to failure in satisfying the real-time requirement. For example, long RTT delay and multiple packet losses that cause slow start phase frequently degrade the end-to-end throughput [29, 30, 31].

Consequently, User Datagram Protocol (UDP) has been considered as an alternate transport protocol for real-time streaming applications. However, UDP does not guarantee reliable data transmission, so that packets can be dropped randomly during transmission. Unreliable service over UDP degrades eventually the quality of the estimated parameters. Another drawback of UDP is that it lacks congestion control and flow control mechanisms, which also invokes the issue of fairness among the connections that share the same bandwidth resources [32].

Currently, most streaming data of the real-time applications are transmitted over UDP through the Internet. As a result, the application programs require additional functions of the transport layer protocol, such as the guarantee of delivery and the congestion control. As an example, Real-time Transport Protocol (RTP) and its companion protocol Real-time Transport Control Protocol (RTCP), which enhance the guarantee of the data delivery by retransmission and the congestion control providing a feedback to source, are being adopted for real-time multimedia applications [33].

A user payload transmitted to the DRS client is either a ray header or a range sample data set. When the DRS server transmits a range sample data set to the client, the server fragments a range sample data set into a couple of datagram packets because the size of a range sample data set is usually larger than that of the typical MTU of 1500 bytes. For the transmission over UDP, the packet loss can occur randomly during transmission because of unpredictable network conditions and insufficiency of buffer space at the client end system [34]. The packet loss causes random loss of the ray headers or the range sample data sets. Loss of the ray headers results in missing rays in the user display. In the same way, loss of the range sample data sets results in the decrease in the number of data sets involved in the estimation of parameters. This degrades the the accuracy of the estimated radar parameters, which results in serious problems in further applications.

The preliminary design of the DRS data transmission over UDP is described in [35, 36]. A server controls the transmission rate by varying the time gap between the transmissions of consecutive range sample data sets after receiving feedback from the client. An inherent issue presented here is that the overall throughput is lower than the data rate from the receiver because of longer time gap so that the server cannot all range sample data sets in a ray DRS block. In such a case, the latter parts of the range sample data sets in sequence number among a ray DRS

block might be dropped at the server. In addition, under more severe congestion some ray DRS blocks might be dropped. These lead to severe degradation in quality of the estimated radar parameters. Further development in maximizing the quality of the end-products under the congestion are described in [37, 38].

In this chapter, we develop transmission waveform designs for congestion control to adapt the transmission rate to the available bandwidth over UDP transport protocol. The server controls the transmission rate based on the feedback from the client. This algorithm employs data selection schemes to provide the highest quality of the estimated radar parameters possible under unpredictable network conditions. The selection schemes are developed relying on the operating principle of the dual-polarized Doppler radar and the theory of polarimetric parameter estimation. The performance of this transmission waveform design scheme combined with the congestion control algorithm is evaluated operating the CSU-CHILL radar in real-time.

4.2 Establishment of QoS for radar data transmission

4.2.1 Theoretical approach

As described in Section 3.2, the CSU-CHILL radar typically transmits a high power pulse (approximately 1 MW) and receives range samples over 100 - 150 km at high resolution such as 150 meters. This is referred to as a range sample data set. At a pointing angle, the pulses are transmitted multiple times to obtain multiple returns from the same observation volume. A collection of the multiple range sample data sets is referred to as a ray DRS block as shown in Fig. 3.1

Table 4.1 presents the dual-polarization Doppler radar parameters that are typically used for the weather radar applications. The parameters are calculated from the estimates of the power averages and covariance elements with lag 0, lag 1, and lag 2. The standard deviations of the estimated parameters essentially control

the quality of subsequent applications, such as rainfall rate estimation, meteorological event classification or severe storm warning. The standard deviations of the radar parameters are affected by a combination of the number of signal samples, signal spectral width, correlation coefficient between polarizations, and signal to noise ratio (SNR) [12, 39]. Some of these factors are determined by the radar operations whereas the rests are determined by prevailing atmospheric conditions.

Parameters	Estimates of power and covariance matrix elements	
	Alternating mode	Hybrid mode
Z	$\widehat{P}_{co}^{v}$	$\widehat{P}_{h}$
Z_{dr}	$\widehat{P}_{co}^{v}$, $\widehat{P}_{co}^{h}$	$\widehat{P}_{v}$, $\widehat{P}_{h}$
LDR_{hv}	$\widehat{P}_{co}^{v}$, $\widehat{P}_{cx}$	
LDR_{vh}	$\widehat{P}_{co}^{h}$, $\widehat{P}_{cx}$	
Velocity	$\widehat{R}_{hh,vv}[1]$, $\widehat{R}_{vv,hh}[1]$	$\widehat{R}_{hh}[1]$
Ψ_{dp}	$\widehat{R}_{hh,vv}[1]$, $\widehat{R}_{vv,hh}[1]$	$\widehat{R}_{vh}[0]$
ρ_{hv}	$\widehat{P}_{co}^{v}$, $\widehat{P}_{co}^{h}$, $\widehat{R}_{hh,vv}[1]$, $\widehat{R}_{vv}[2]$	$\widehat{R}_{hh}[1]$, $\widehat{P}_{h}$
Spectral width		$\widehat{R}_{vh}[0]$, $\widehat{R}_{hh}[1]$

Table 4.1: Radar parameters.

Moreover, when the DRS is transferred over a data network like the high-bandwidth VCHILL, the network condition becomes an additional factor influencing the estimation of the radar parameters. For example, when the real-time DRS is transferred to remote sites over UDP, packet losses can happen depending on the network traffic condition and available buffer size of the client end system. Since the variance of parameter is inversely proportional to the square of the number of samples that are involved in the estimation, the availability of the range sample data sets is critical in decreasing the variances [12, 39]. Another important factor is that some of radar parameters, such as velocity, Ψ_{dp}, ρ_{hv}, and spectral width, are calculated with the estimators of correlation and covariance as shown in Table 4.1. The random loss of the packets causes the deficiency of the consecutive time

samples and associated estimators at the remote sites, which produces no corresponding parameters. Therefore, it is necessary to develop an transport protocol in the application layer to solve these problems.

4.2.2 Simulation study

To study the effect of the data-loss pattern on the average and standard deviation of the estimated radar parameters, three scenarios of the data-loss were simulated assuming the real-time transmission over UDP. For the first scenario, the server transmits the DRS without any congestion control. The transmission rate is constant at a data rate irrespective of the network traffic condition so that the data loss occurs randomly. For the second scenario, the server transmits the DRS while operating a rate control algorithm in which the time gap between the data transmissions is varied depending on the network condition [35, 36].

For the third scenario, the server transmits the DRS while operating the rate control algorithm proposed in this chapter. The DRS server deliberately drops some range sample data sets and transmits selected data to adapt the transmission rate to the available bandwidth so that the transmission rate is controlled. This is equivalent to synthetic waveform design by suppressing transmit pulses. Figure 4.1 and 4.2 provide the proposed schemes for selecting the range sample data sets to be transmitted depending on the available bandwidth for the alternating mode and the hybrid mode, respectively. The server selects paired range sample data sets so that the selected pairs are evenly distributed along the row of a ray data block, based on the specified rules described as belows. For the alternating mode, the parameters, such as velocity, Ψ_{dp}, and ρ_{hv}, are calculated from the correlation estimators of lag 0, lag 1, and lag 2. Therefore, at least three consecutive range sample data sets are required to estimate those parameters. For the hybrid mode, the parameters, such as velocity, Ψ_{dp}, ρ_{hv}, and spectral width, are calculated from

the correlation estimators of lag 0 and lag 1. Hence, at least two consecutive range sample data sets are required to estimate those parameters.

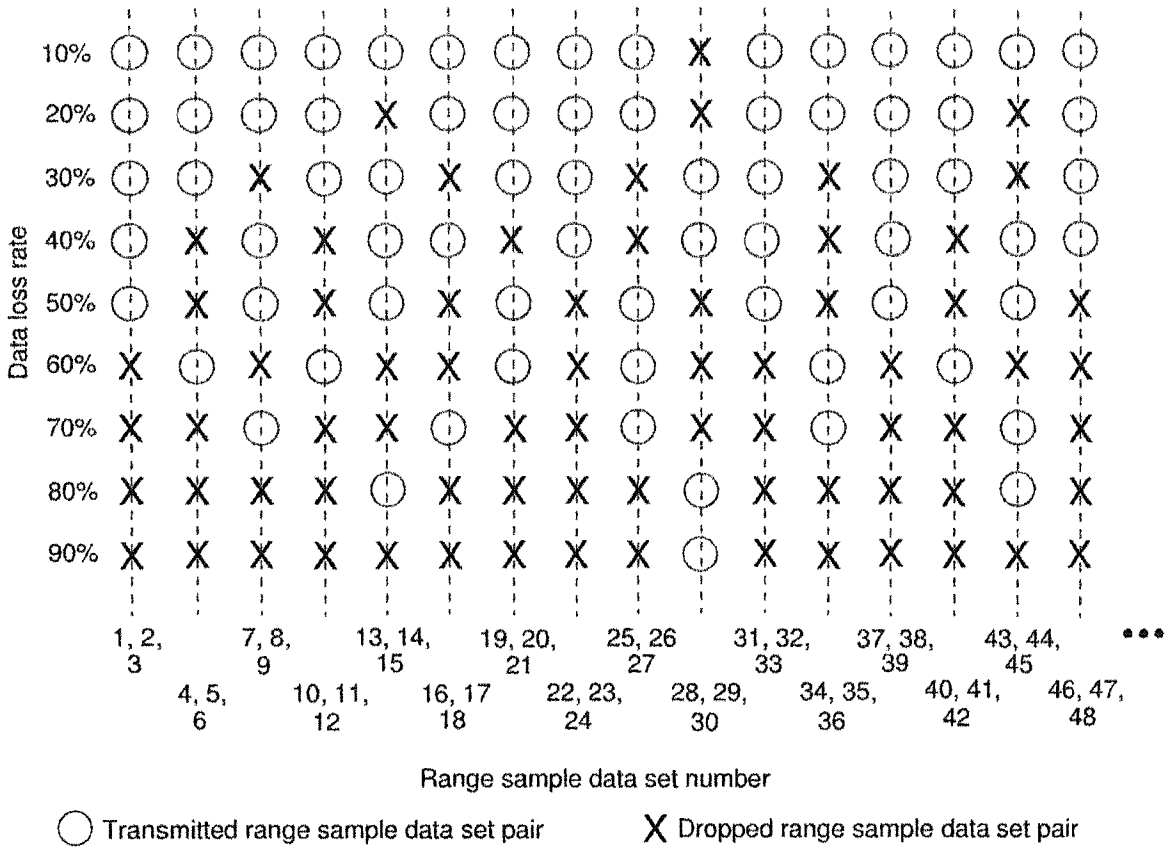


Figure 4.1: Synthetic waveform design scheme depending on the data loss rate for the alternating mode.

Note that the available bandwidth is associated with the data-loss rate once the data rate is set up. Data rate generated by a radar is determined by a combination of various factors, such as quantization level, receiver bandwidth, and number of receive channels. For a coherent receiver of dual channels with the sampling rate of 1 MHz and the quantization level of 16 bits, the data rate is estimated to be approximately 64 Mbps. Therefore, if the available bandwidth is 64 Mbps, no transmitted data would be lost. However, if the available bandwidth is 32 Mbps, then 50% of the transmitted data would be lost. The data-loss rates

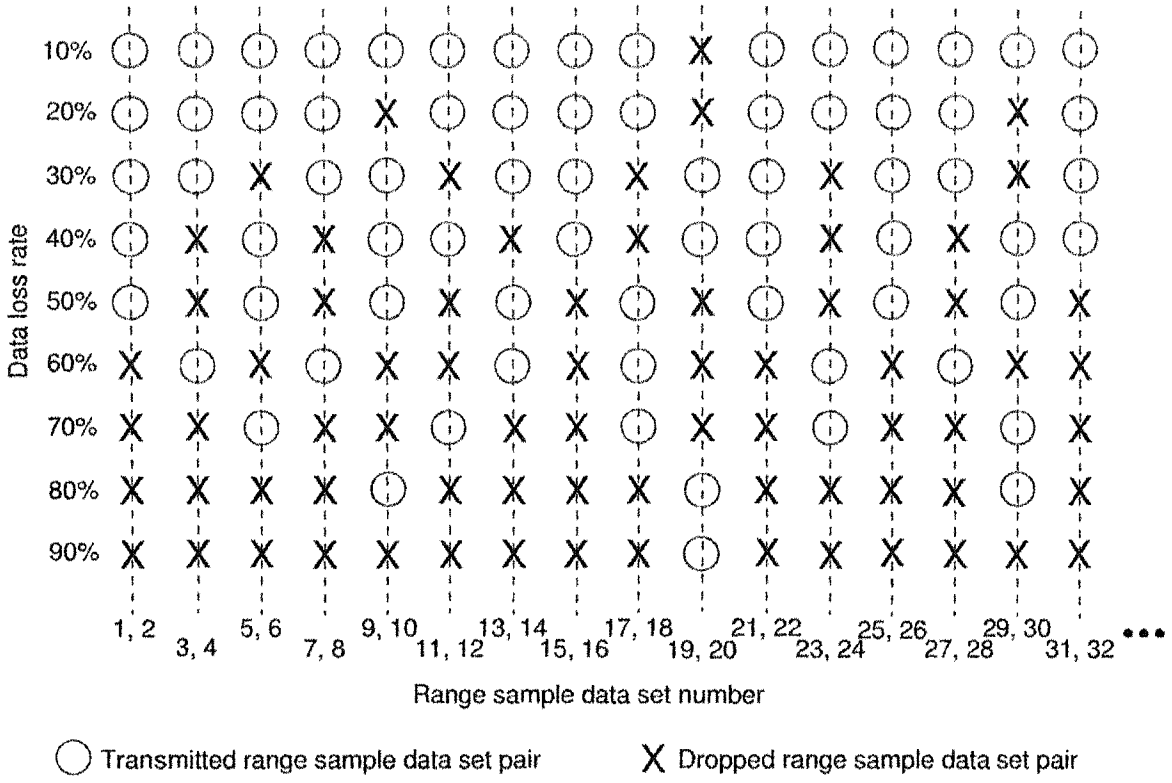


Figure 4.2: Synthetic waveform design scheme depending on the data loss rate for the hybrid mode.

(available bandwidths) are ranked into 10 levels, ranging from no data loss to 90% of data-loss rate as illustrated in Fig. 4.1 and 4.2.

For this study, simulated DRS with the hybrid-mode operation were used because of the ease with evaluating the effect of the data loss pattern on the estimation of parameters. The validity of the DRS data simulation and the estimating algorithms was confirmed in advance of the study. Figure 4.3 shows the confirmation procedure. The DRS were simulated using various data sets of the parameters that are shown in Table 4.2 [40, 41, 42, 12]. Subsequently, the radar parameters were estimated with the simulated DRS using the algorithm as described in Appendix C. The mean and standard deviation of the estimated radar parameters were calculated over the 1000 gates. Finally, the means of estimated parameters were compared to the input parameters.

Figure 4.3: Procedure to confirm the DRS simulation and the radar signal estimation algorithms.

Reflectivity (dBZ)	Z_{dr} (dB)	ρ_{hv}	*phi* (degree)	Mean velocity (m/sec)	Spectral width (m/sec)	SNR (dB)
10	3	1	45	10	3	10
10	3	1	45	10	3	15
10	3	1	45	10	3	20
10	3	1	45	10	3	30

Table 4.2: Dual-polarization Doppler radar parameters used for simulating the DRS. The radar operation is assumed to be in hybrid mode.

Figure 4.4, 4.5, 4.6, and 4.7 show the changes in standard deviations of the estimated parameters varying the data-loss rate. Simulation results show that even though the mean of the estimated radar parameters does not change much, the standard deviations increase proportional to the data-loss rate for all scenarios. In particular, the third scenario, in which the transmission wavefrom design scheme is employed, shows the lowest increase in the standard deviation when the data-loss rate is below 80%. The second scenario shows the highest increase in standard deviation.

The simulation study shows that the standard deviation of the estimated radar parameters is a suitable quality of service (QoS) for the performance evaluation of the high-bandwidth VCHILL. In addition, the simulation study presents that the optimized selection of data to be sent can provide the high-quality radar parameters. A combination of the transmission waveform design scheme and the congestion control algorithm, which provides the highest quality of service possible over unpredictable network conditions, will be described in the next section.

4.3 Congestion control algorithm

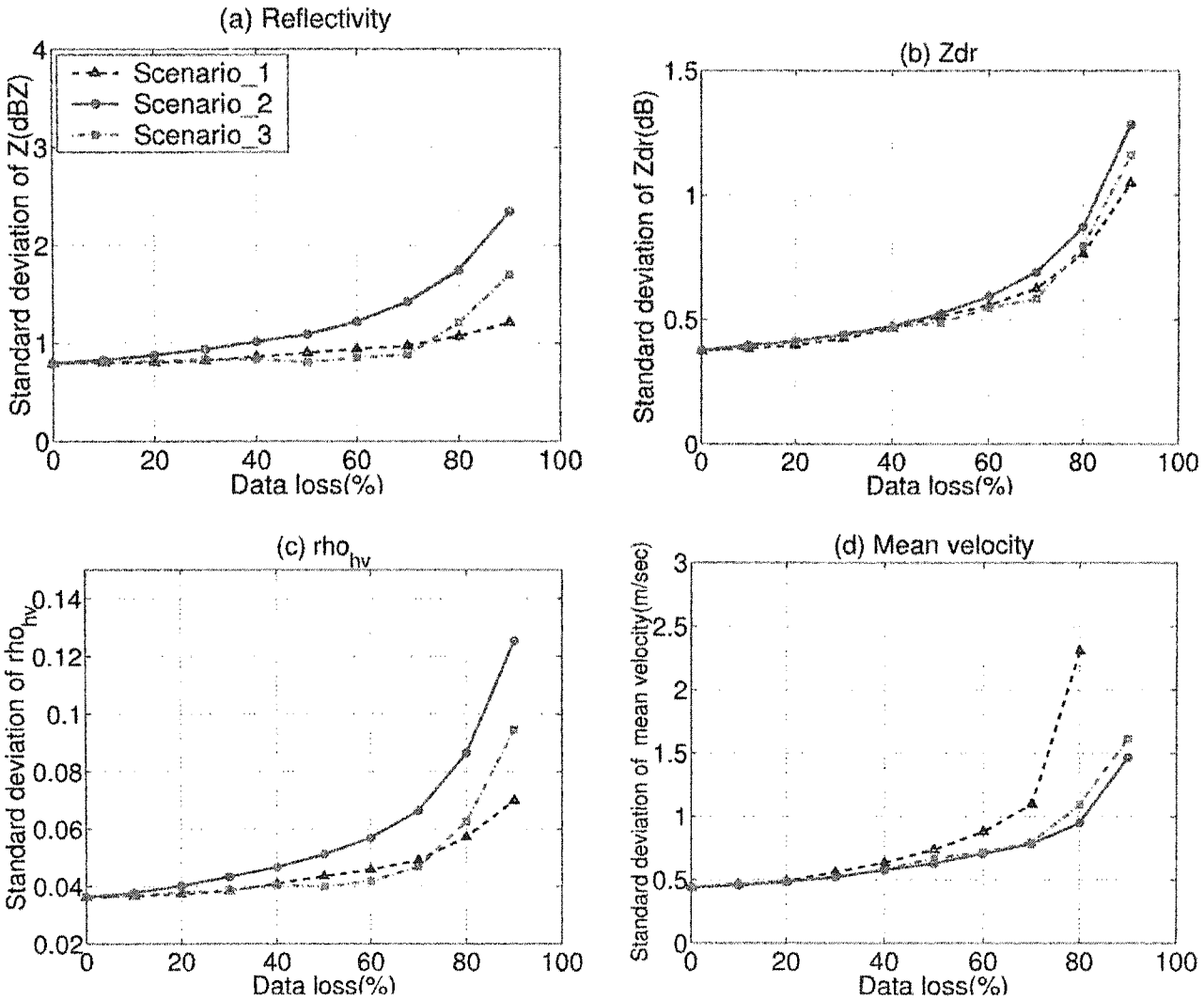


Figure 4.4: Changes in standard deviation of estimated radar parameters varying the data-loss scenario and the data-loss rate. Scenario 1 is the case where the data are lost randomly, scenario 2 is the case where the last parts of the data in a ray are lost, and scenario 3 is the case where the data are intelligently dropped at a server as shown in Fig. 4. SNR is 10 dB. (a) Reflectivity (b) Z_{dr} (c) rho_{hv} (d) mean velocity

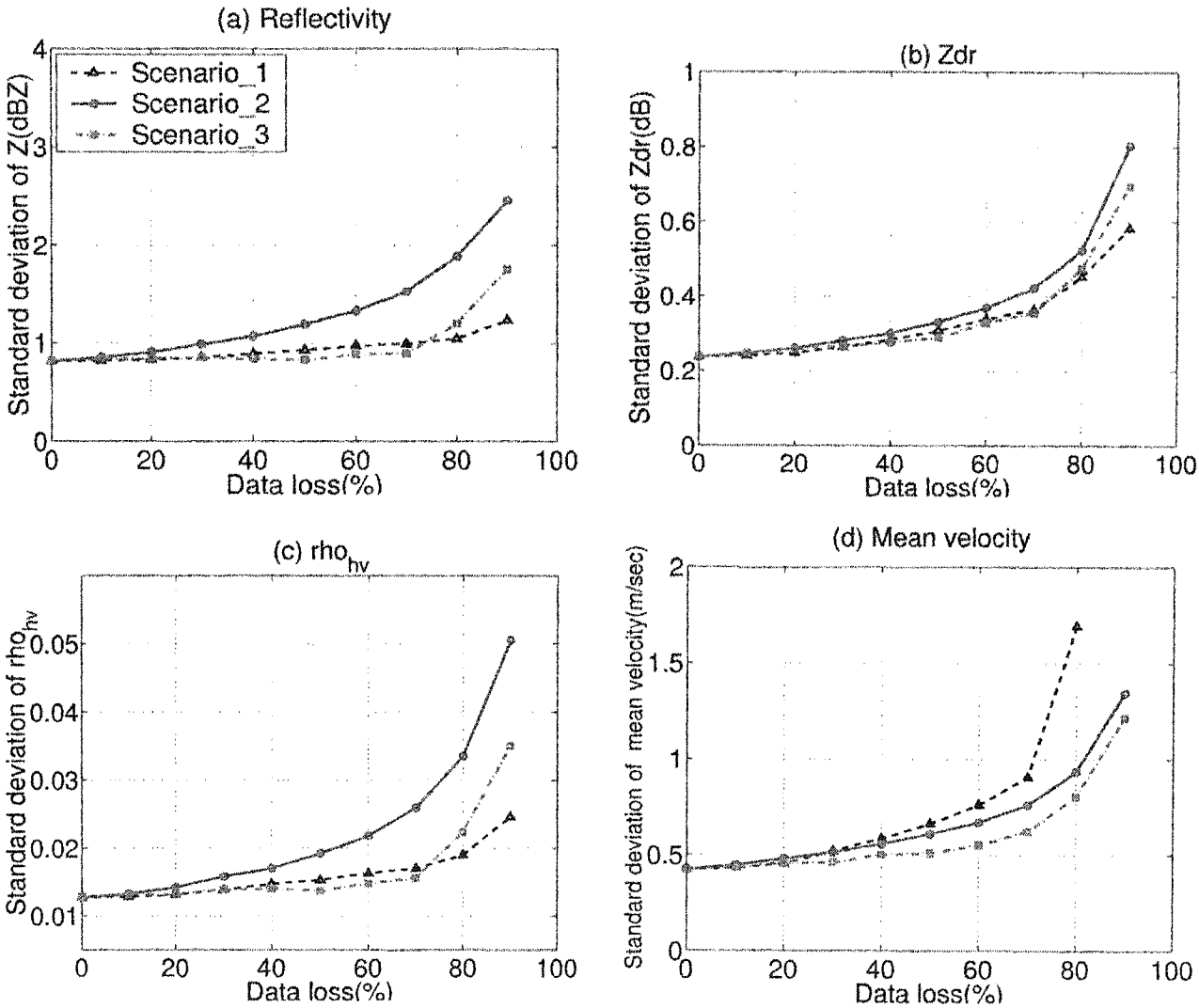


Figure 4.5: Changes in standard deviation of estimated radar parameters varying the data-loss scenario and the data-loss rate. Scenario 1 is the case where the data are lost randomly, scenario 2 is the case where the last parts of the data in a ray are lost, and scenario 3 is the case where the data are intelligently dropped at a server as shown in Fig. 4. SNR is 15 dB. (a) Reflectivity (b) Z_{dr} (c) rho_{hv} (d) mean velocity

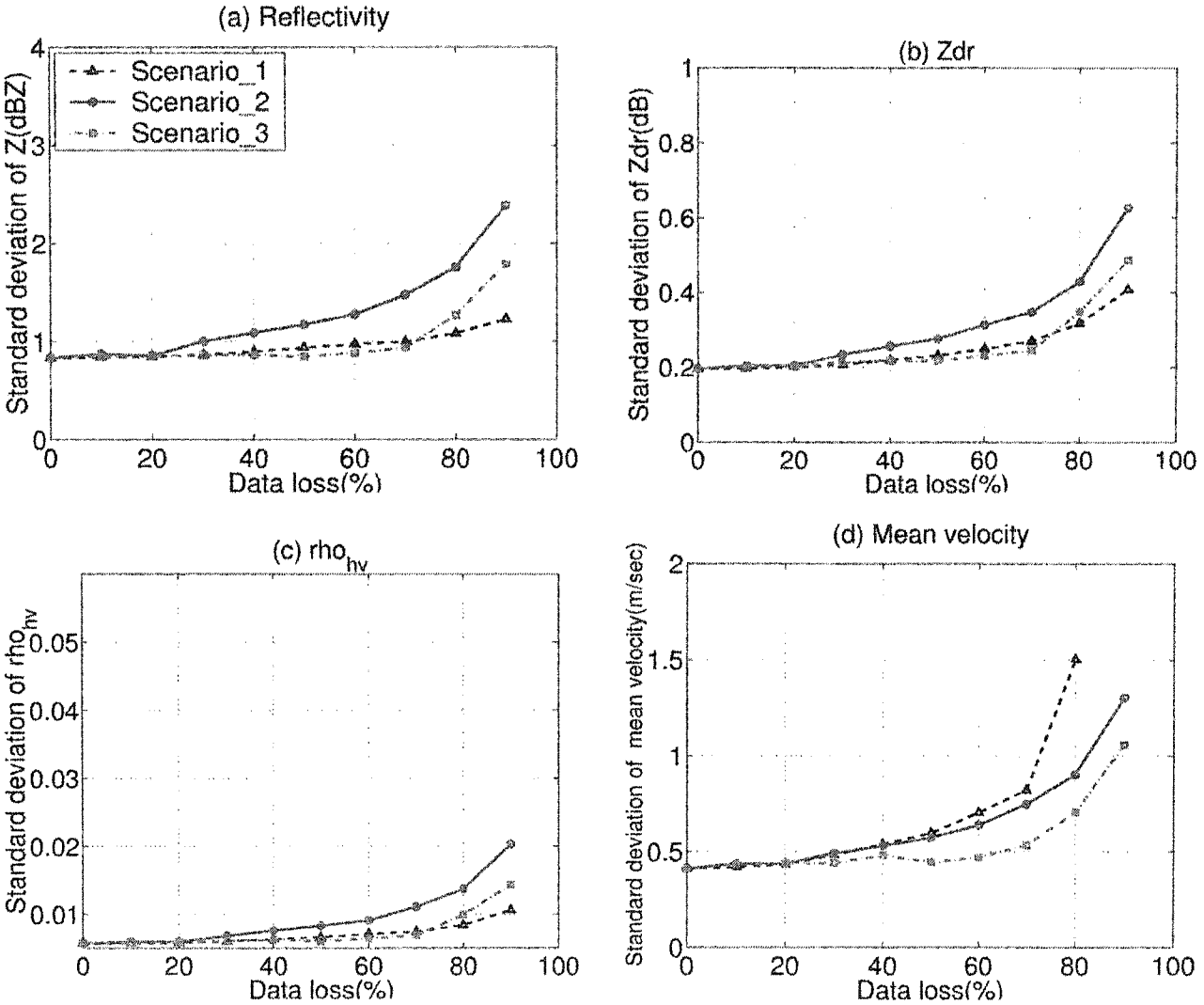


Figure 4.6: Changes in standard deviation of estimated radar parameters varying the data-loss scenario and the data-loss rate. Scenario 1 is the case where the data are lost randomly, scenario 2 is the case where the last parts of the data in a ray are lost, and scenario 3 is the case where the data are intelligently dropped at a server as shown in Fig. 2.1. SNR is 20 dB. (a) Reflectivity (b) Z_{dr} (c) rho_{hv} (d) mean velocity

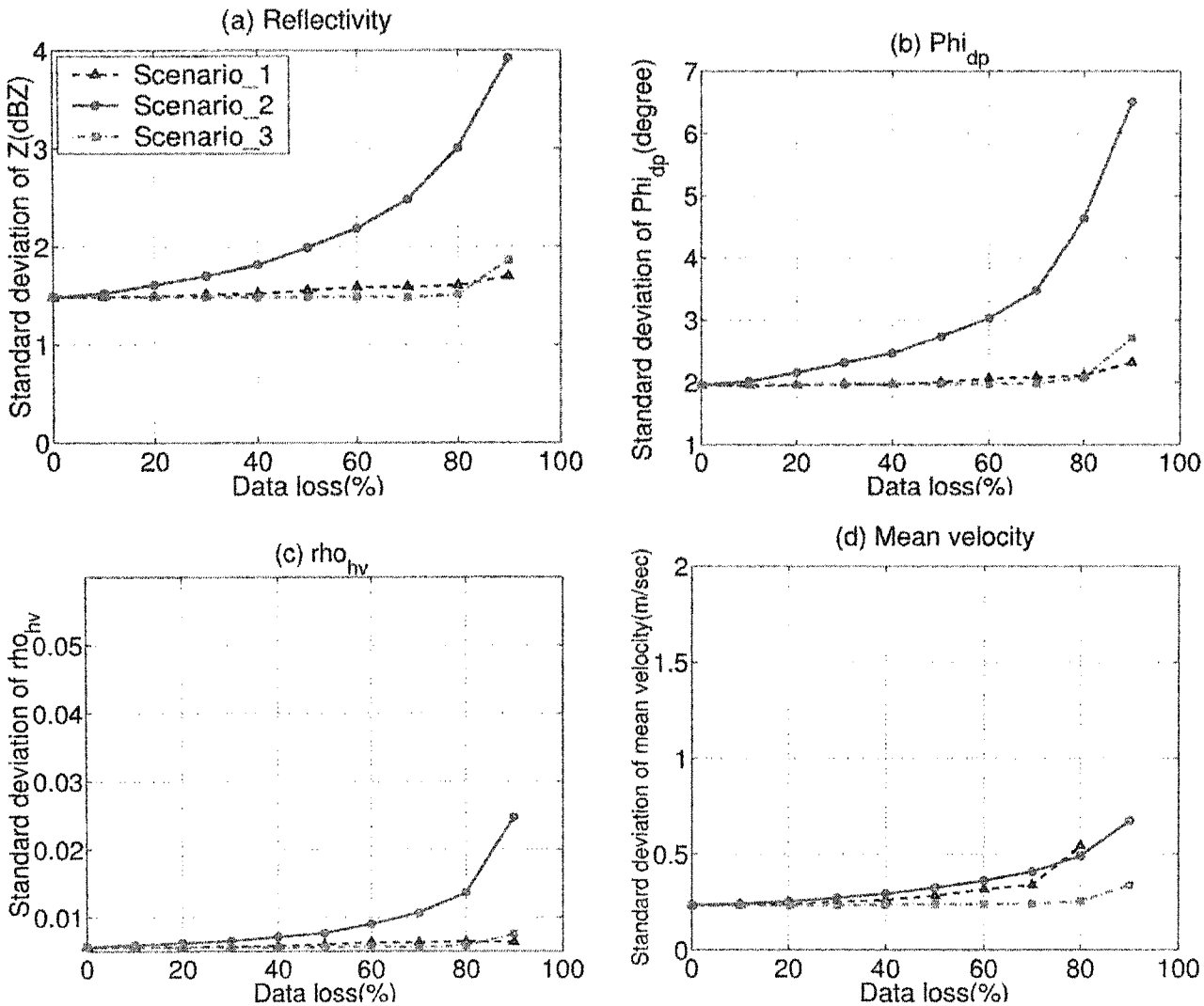


Figure 4.7: Changes in standard deviation of estimated radar parameters varying the data-loss scenario and the data-loss rate. Scenario 1 is the case where the data are lost randomly, scenario 2 is the case where the last parts of the data in a ray are lost, and scenario 3 is the case where the data are intelligently dropped at a server as shown in Fig. 2.1. SNR is 30 dB. (a) Reflectivity (b) Z_{dr} (c) rho_{hv} (d) mean velocity

4.3.1 Congestion control algorithm

The proposed congestion control algorithm is classified as the source-based rate control with the Additive Increase/Loss Proportionate Decrease (AIPD) algorithm [28, 43]. The server estimates available bandwidth based on the feedback from the client and controls the transmission rate. The client measures the amount of data loss, and sends feedback including the transmission level and the amount of data loss. The transmission rate is updated every time just after completing the transmission of a ray DRS block, and lasts during transmission of next ray DRS block.

The transmission level is defined such that the data rate is divided into 10 levels. Transmission level 10, which is the maximum transmission rate (R_{max_trans}), indicates entire data transmission at the data rate whereas level 1 corresponds to transmission with 10% of the data rate. Each transmission level also corresponds to a data-loss rate, as explained earlier. Table 4.3 shows the relation between the transmission level, transmission rate, data-loss rate, and percentage of the transmitted data.

The ray header includes the transmission level and the total number of range sample data sets, which enables the DRS client to expect which range sample data sets in a ray DRS block will be delivered. The expected range sample data sets are predefined according to the Fig. 4.1 and 4.2 as a protocol between the DRS server and client. Based on this expectation, the client counts the amount of lost range sample data sets during the transmission of a ray DRS block. The lost amount is calculated by subtracting the amount of delivered data from the amount of expected data. Note that when all expected amount of the data at a specific transmission level arrives at the client, then the number of lost data is counted as zero. The feedback packet contains the transmission level and the

Transmission level	Transmission rate (Mbps)	Data-loss rate (%)	Percentage of data transmitted (%)
10	R_{max_trans}	0	all data generated
9	$\frac{9}{10}R_{max\_trans}$	10	$\frac{9}{10}$
8	$\frac{8}{10}R_{max\_trans}$	20	$\frac{8}{10}$
7	$\frac{7}{10}R_{max\_trans}$	30	$\frac{7}{10}$
6	$\frac{6}{10}R_{max\_trans}$	40	$\frac{6}{10}$
5	$\frac{5}{10}R_{max\_trans}$	50	$\frac{5}{10}$
4	$\frac{4}{10}R_{max\_trans}$	60	$\frac{4}{10}$
3	$\frac{3}{10}R_{max\_trans}$	70	$\frac{3}{10}$
2	$\frac{2}{10}R_{max\_trans}$	80	$\frac{2}{10}$
1	$\frac{1}{10}R_{max\_trans}$	90	$\frac{1}{10}$

Table 4.3: Each transmission level is associated with a transmission rate, data-loss rate and percentage of data transmitted.

amount of lost data in a ray DRS block as shown in Table 4.4. The server updates the transmission rate upon the arrival of the feedback packet.

If no data loss is reported, the server increments the transmission level by one. However, if some amount of data loss is reported, the server adapts the transmission rate to the available bandwidth. The server calculates the total amount of lost data for the transmission with respect to the total amount of data generated at the radar. According to this updated transmission rate, the range sample data

Header ID	0 - request for transmission, 1 - feedback, 2 - request for retransmission
Message	Reserved
Sweep number	Integer
Ray number	Integer
Transmission level	1 - 10
Data loss	Total amount of data loss at a transmission level
Data number for retransmission	Reserved

Table 4.4: Structure of feedback packet. Retransmission associated fields are included for future upgrade.

sets to be sent are selected depending on the radar operation mode as shown in Fig. 4.1 or 4.2.

4.3.2 Overall timing sequence

Figure 4.8 shows a timing chart of data exchange between the DRS server and client. All data including the transmission of DRS and feedback are exchanged over UDP. Initially, the server transmits a ray header after getting a request of the DRS data transmission. Subsequently, the server starts to transmit the range sample data sets. The client sends the feedback packet just after the arrival of the last range sample data set of a ray. The feedback packet may arrive at the server during the transmission of following ray DRS block because of the propagation delay. The server checks the arrival of the feedback packet after transmitting the last data of a ray. Note that the server UDP socket is set as a nonblocking I/O mode [23]. Therefore, once the server finds a feedback packet to read, the server reads it, estimates the available bandwidth, and updates the next sending rate. However, if no feedback packet is found in the socket buffer, then the server continues to transmit the next ray DRS block at the previous transmission rate. A time lag, which is at least the time taken for the transmission of a ray DRS block, exists between transmitting the last range sample data set and reading the corresponding feedback. Depending on the propagation delay, the time lag can be extended. For example, the server transmits the first and the second ray DRS block with a default transmission rate in Fig. 4.8. During the transmission of the second ray DRS block the feedback packet of the first ray transmission arrives at the server. Then, the server reads this feedback after completing transmission of the second ray DRS block, updates the transmission rate, and applies it to transmission of the third ray DRS block.

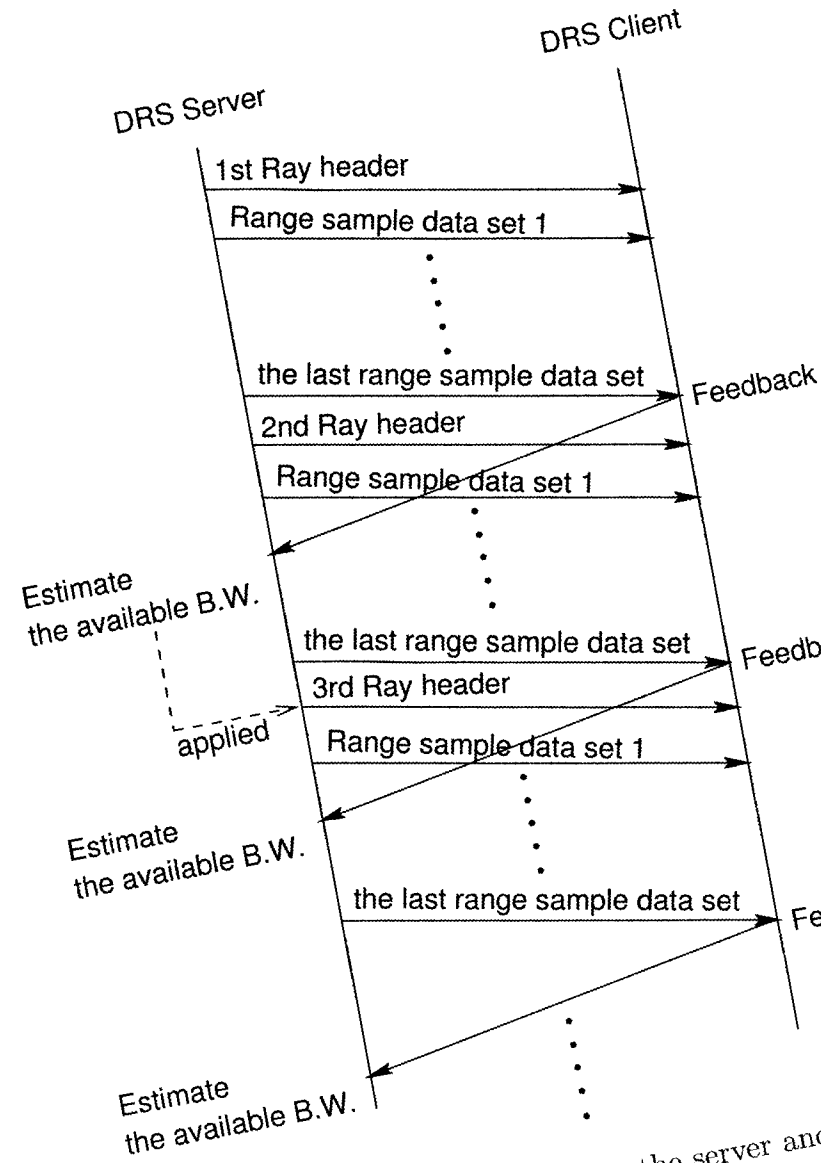


Figure 4.8: Timing diagram between the server and the client.

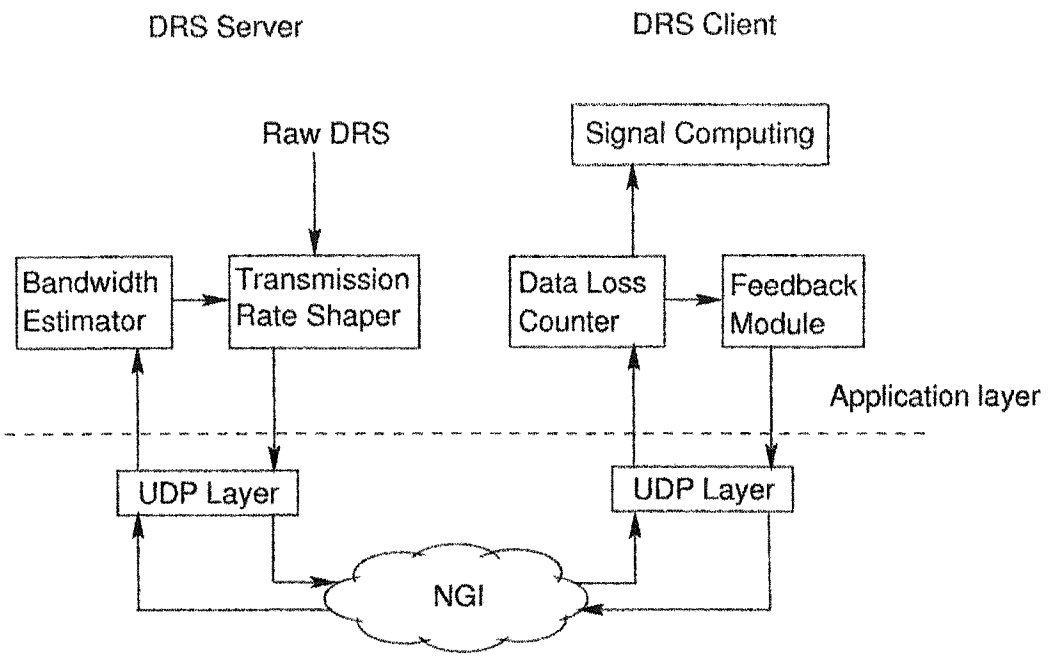


Figure 4.9: An end-system architecture for performing congestion control over UDP.

4.4 Implementation and performance evaluation

Figure 4.9 provides the end-system architecture for the congestion control combined with the transmission waveform design scheme. The existing application programs designed over TCP was modified to include the proposed congestion control algorithm. The service thread of the DRS transmission process on the DRS server and the DRS receive thread of the DRS receive process on the DRS client, which are responsible for the transmission of data over the data network, were mainly subject to the modification. On the DRS server, the bandwidth estimator calculates the data-loss rate and estimates the available bandwidth. The bandwidth estimator passes the information to the transmission-rate shaper that selects the range sample data sets to transmit among a ray DRS block. On the client, the data-loss counter counts the amount of lost data, and the feedback module sends the relevant information to the server.

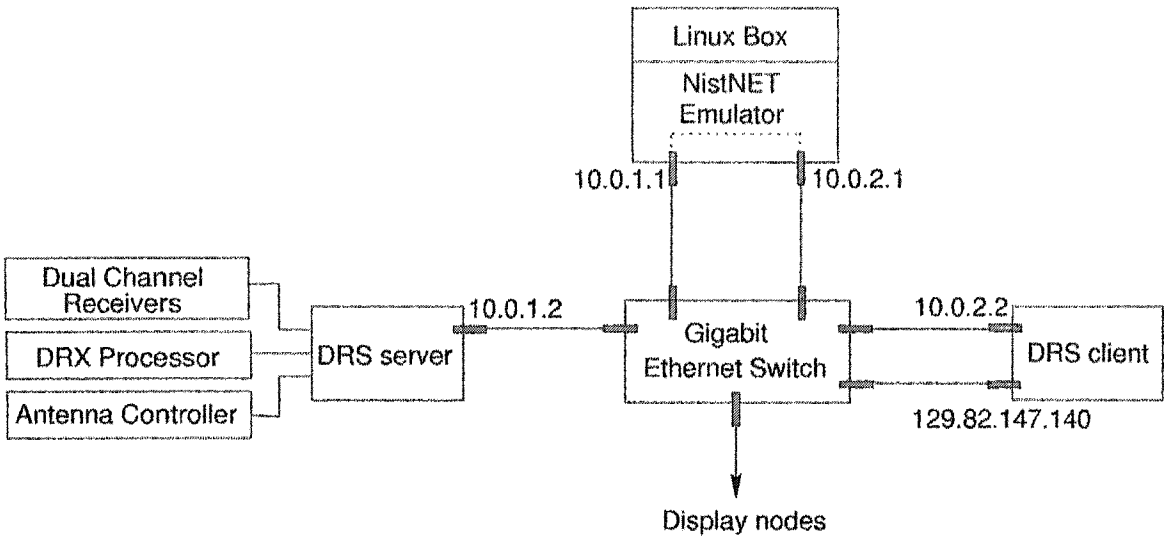


Figure 4.10: Test bed for the performance evaluation of the the implemented congestion control algorithm.

Our proposed congestion control algorithm was evaluated in real-time using a simple test bed, while running the CSU-CHILL radar. The test bed consists of the DRS server and client connected to each other through a network emulator over a gigabit link as shown in Fig. 4.10. NIST Net emulation package (Version 2.0.12) changes the available bandwidth of the established gigabit link between the server and the client [44]. One-way delay was set to be 25 msec. The important operating conditions of CSU-CHILL radar is presented in Table 4.5. The transmission level, which is only effective during transmission of a ray DRS block, was recorded varying the available bandwidth. Figure 4.11 shows the change in the transmission level for 10 minutes varying the available bandwidth. The DRS server dynamically adapts the transmission rate to the available bandwidth. Figure 4.12 shows the improvement of the end user display by the deployment of the congestion control. These figures were captured from a display window showing the reflectivity just after the second sweep display was completed.

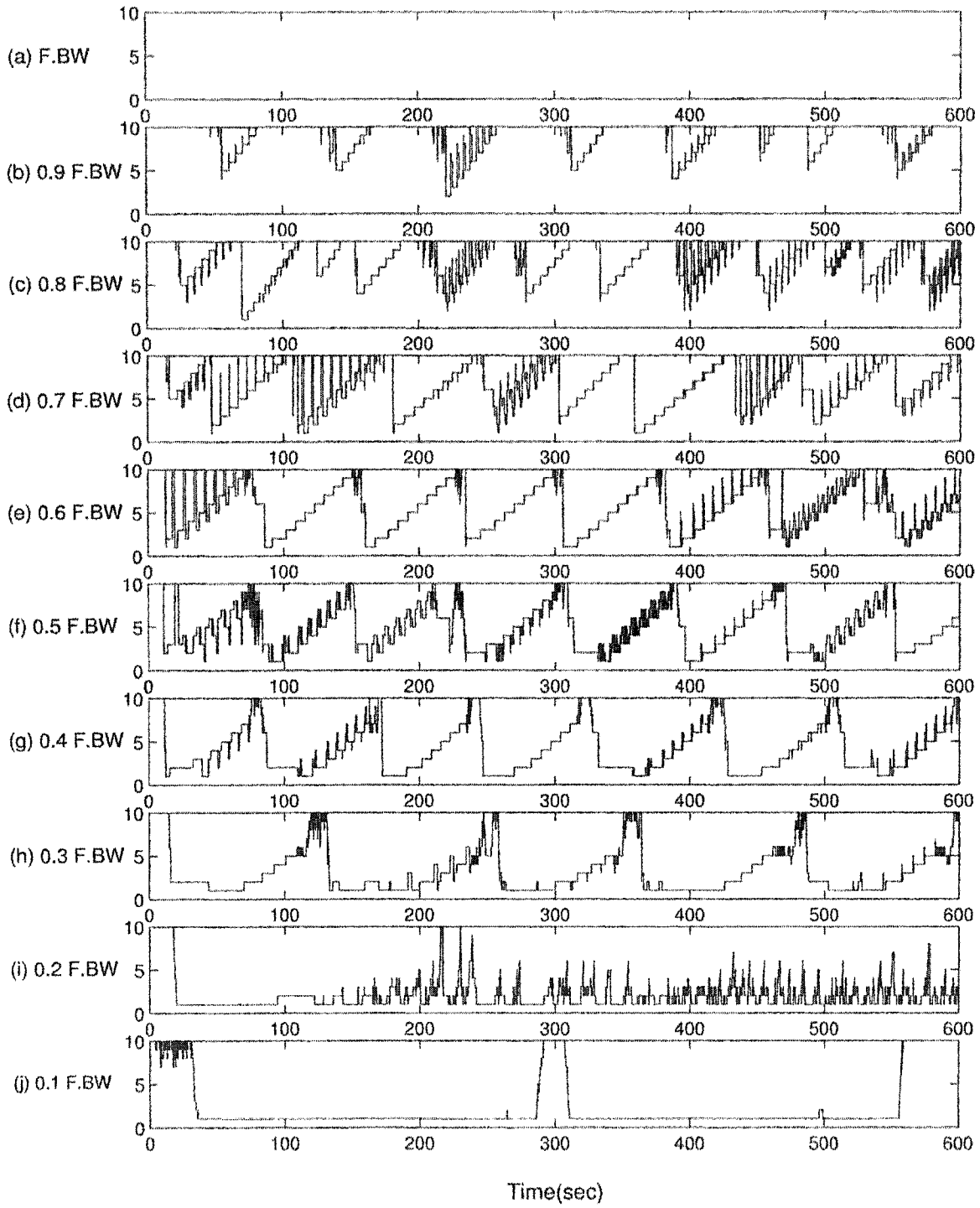


Figure 4.11: The transmission level changes according to the available link bandwidth. The X axis and Y axis are time and transmission level, respectively. The measurements were conducted for 600 sec, in which 10 sweep data were transmitted. The F. BW stands for full required bandwidth for the data rate of 64 Mbps.

Pulse repetition time	1 msec
Number of pulses emitted for a ray	128
Angular resolution of a ray	0.75 degree
Sampling rate	1 MHz

Table 4.5: Operating conditions of CSU-CHILL radar for the performance evaluation of the implemented congestion control algorithm.

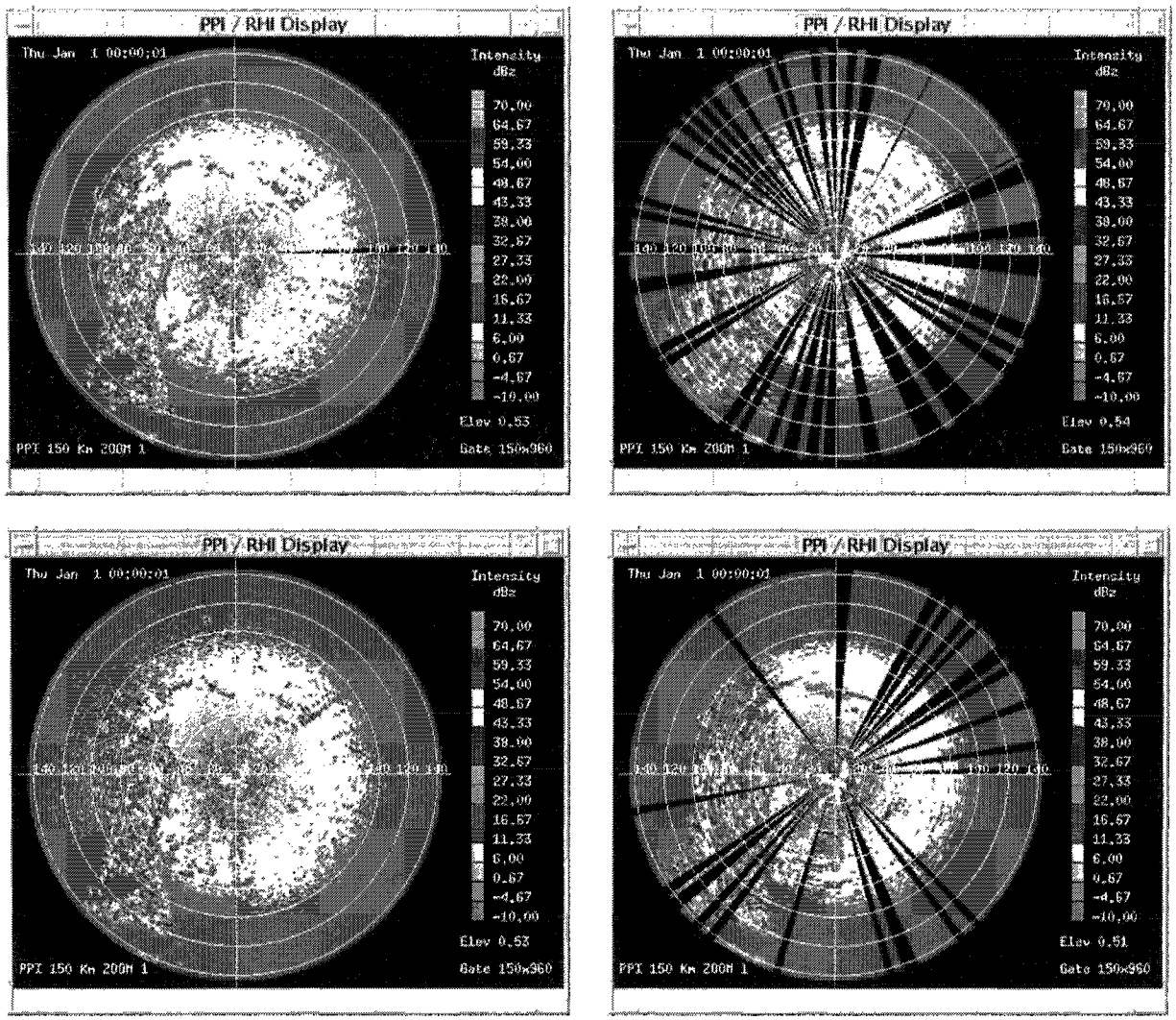


Figure 4.12: Comparison of displays between the case when our proposed algorithm is conducted (left column) and the case when no congestion control is applied (right column). The available bandwidths of the first row and the second row are 0.7 F.BW and 0.6 F.BW, respectively.

4.5 Discussion

The use of the signal driven I/O on the server is an option for the socket to receive the UDP packet sent from the client [23]. An advantage of this option is that the application program is not blocked while waiting for the datagram to arrive. The main loop can continue executing and just wait to be notified by the kernel that the datagram is ready to read. This socket option allows the client to send various kinds of packets to the server, such as feedback and retransmission request when it realizes that some data have been lost. Then, the server can take action depending on the type of the arrived datagram. For instance, if the server receives the request of a retransmission, it sends the lost data again during the transmission of current data. It was confirmed that the signal driven I/O worked with a simple test program. However, it did not work on the real implemented programs for the DRS transmission.

As pointed out earlier, the server sends the ray headers and the range sample data sets. The ray header contains the radar operating condition information required for estimating various radar parameters. Consequently, once a ray header is lost, then the following range sample data sets are useless and cannot be involved in the estimation, which wastes the bandwidth of the data network. Similarly, the feedback packet loss degrades the dynamics in controlling the transmission rate. Therefore, an intelligent way to guarantee the ray header and the feedback delivery is necessary. Although the TCP-based transmission of ray header and feedback packet has been tried, synchronization control in exchanging data over both TCP and UDP was not easy at high transmission rate of a few hundred Mbps.

Moreover, some fine tuning of the algorithm is necessary to be carried out considering the real network traffic feature. For example, the buffer size of the nodes in the network and end node can affect the dynamics of transmission rate control. Further study on this topic is necessary.

4.6 Summary and Conclusion

The quality of the estimated parameters is critical for the weather radar applications. When the DRS are transmitted over UDP, the data might be lost during transmission because of the insufficiency of the network bandwidth and buffer size of the client. This affects the quality of the parameters estimated at the remote sites. A transmission waveform design schemes tailored for the high-bandwidth VCHILL over UDP have been proposed to provide tolerance of the real-time operation and high quality of end products. The waveform design schemes rely on the operating principle of the pulsed-Doppler radar and the estimation theory of the polarimetric radar parameters. The DRS server groups two or three consecutive range sample data sets depending on the radar operation mode, transmits some of the groups, and suppresses other transmissions according to the available bandwidth. Simulation study presents that the proposed data selection scheme is extremely effective in providing high quality end products. The waveform design schemes have been combined with the source-based rate control with a AIPD based on the feedback to develop congestion control algorithm in the high-bandwidth VCHILL over unpredictable network. The performance of the proposed congestion control algorithm was evaluated via real-time operation of CSU-CHILL radar. The results clearly show that the server dynamically adapts the transmission rate to the available bandwidth. The end-user displays also provide much higher quality in comparison with the case with no congestion control.

Chapter 5

DEVELOPMENT OF THE DISTRIBUTED DRS CLIENT

5.1 Introduction

Pulsed-Doppler weather radars average the echo signals from the individually transmitted pulses to reduce the statistical uncertainty of estimates of spectral moments and polarimetric parameters for each resolution volume [12, 39]. For the radar meteorology applications, such as meteorological events classification and rain rate estimation, the quality of the estimated parameters is critical to obtain expected results. For this reason, innovative signal processing algorithms and techniques, such as clutter filtering, spectral processing, and oversampling [10], have been being developed in the weather radar applications.

Advantage of the high-bandwidth VCHILL is that users at the remote sites can receive the DRS in the networked environment and process the DRS for various applications in real-time [37, 38]. For the stable operation of the high-bandwidth VCHILL, network bandwidth and end-system capability should be sustained enough to support the DRS transmission at a data rate. For the DRS transmission over TCP, insufficient computing capability might reduce the end-to-end throughput and lead to failure in satisfying the real-time requirements as described in Section 3.5. Moreover, the implementation of extended signal processing at the remote sites increases the computing workloads much so that an end system might be unable to handle the increased workloads. In some cases, the

implementation of an innovative signal processing increases both the data rate and the computing loads simultaneously. Oversampling radar signals in range time is an example [10]. Since the data rate of the DRS is linearly proportional to the sampling rate assuming that the quantization level and number of channels are unchanged, the oversampling increases the data rate by a specified factor. Moreover, it requires whitening transformation over range samples, which increases the computing workloads at the client.

In this chapter, we develop a concept of distributed DRS client consisting of networked multiple nodes to increase overall computing capability. The essential idea is to distribute the computing workloads among the networked multiple nodes and combine the individual results. The architecture and the scheme to distribute the workloads are designed based on the operating principle of the pulsed-Doppler radar and the theory of polarimetric parameter estimation. The idea is successfully demonstrated by implementing the end systems on Linux boxes in a networked environment and evaluating the end-to-end throughput between the DRS server and the distributed client system.

5.2 Architecture for the distributed DRS client

5.2.1 End-node architecture

Figure 5.1 provides a conceptual diagram of the distributed DRS client. Among the distributed nodes data are exchanged over TCP through high speed networks, namely Gigabit Ethernet or Myrinet [45]. Moreover, another library specifications, such as Message Passing Interface (MPI) [46], can be used for clustering workstation-based end systems. A node serves as a master node, which is responsible for communication with the DRS server, distribution of the received DRS data, integration of the radar parameters, and transmission of the complete set of parameters. It can be also involved in the estimation of radar parameters partially.

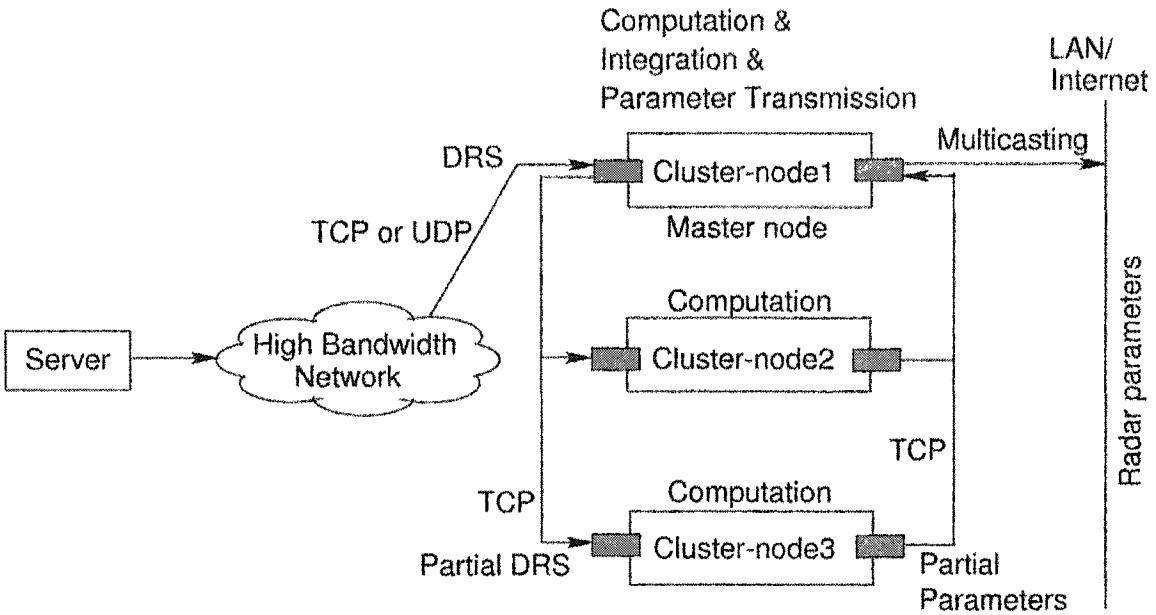


Figure 5.1: Conceptual diagram for the distributed DRS node to increase the computing capability.

Signal processing on the DRS are mostly applied to the time samples associated with a range and the parameters are estimated on the range. For the whitening transformation, the transformation is applied to the oversampled consecutive range samples and subsequently applied to the sampled data sets. Therefore, the ray DRS block which is visualized as a two-dimensional array can be partitioned into multiple data groups along the columns, and then the data groups are distributed among the nodes as shown in Fig. 5.2. Note that the workloads should be distributed evenly among all nodes to achieve the highest performance possible without any bottleneck in processing and estimation.

Figure 5.3 illustrates the network configuration and the end-system architecture for the distributed DRS client. The DRS server operates with the same architecture as that described in Chapter 3. Each node of the distributed DRS client includes the DRS receive process and the parameter transmission process. In the master node, the DRS receive process consists of a receive/retransmit thread and a parameter computing thread. The parameter transmission process is composed of

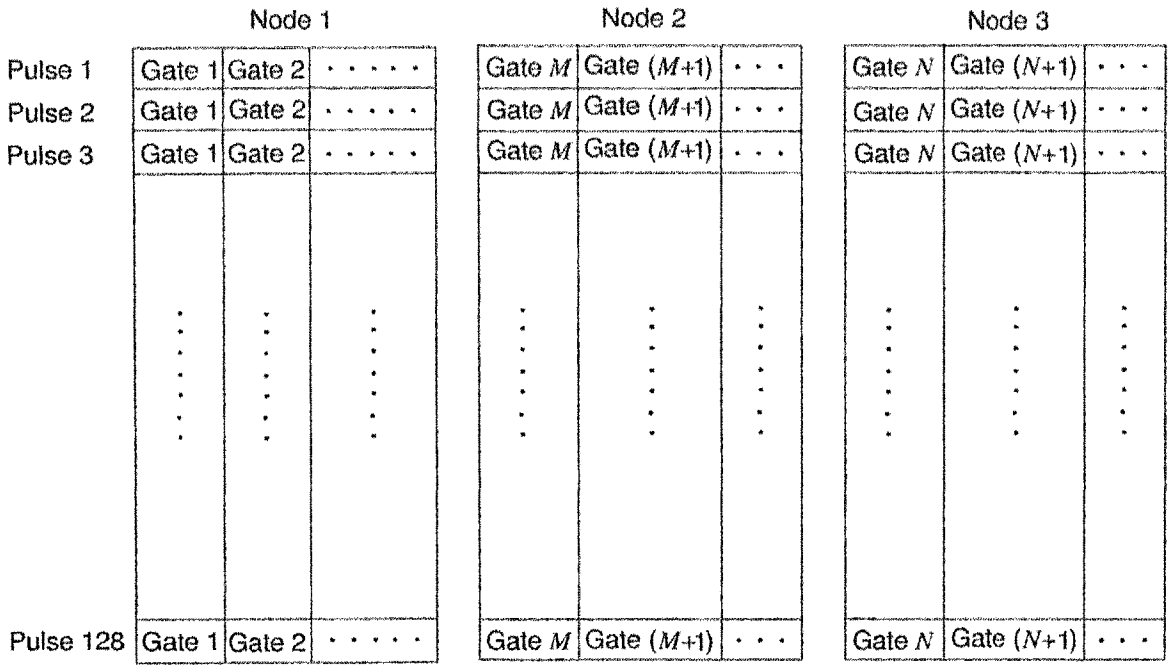


Figure 5.2: A ray DRS block is divided into multiple groups to be distributed to other DRS client nodes. Here, a ray DRS block with 128 range-sample data sets is divided into 3 groups.

integration thread and parameter transmission thread. However, the other nodes have the same architecture with that of the DRS client described in Chapter 3 except the transport protocol for the parameter transmission. In this configuration of the distributed DRS client, all data are exchanged over TCP.

Two TCP sockets in the receive/retransmit thread, one is for connecting to the DRS server as a client and another is for retransmitting the partial DRS to other distributed nodes as a server, are created. In the beginning of operation, the master node connects to the DRS server via the client TCP socket and then waits until other distributed nodes establish the connection to it via the server TCP socket. After establishing the connections between the master node and the distributed nodes, the master node sends the request for the DRS transmission to the DRS server. The receive/retransmit thread in the master node distributes the received DRS data to the distributed nodes and passes partial DRS to the parameter computing thread of itself.

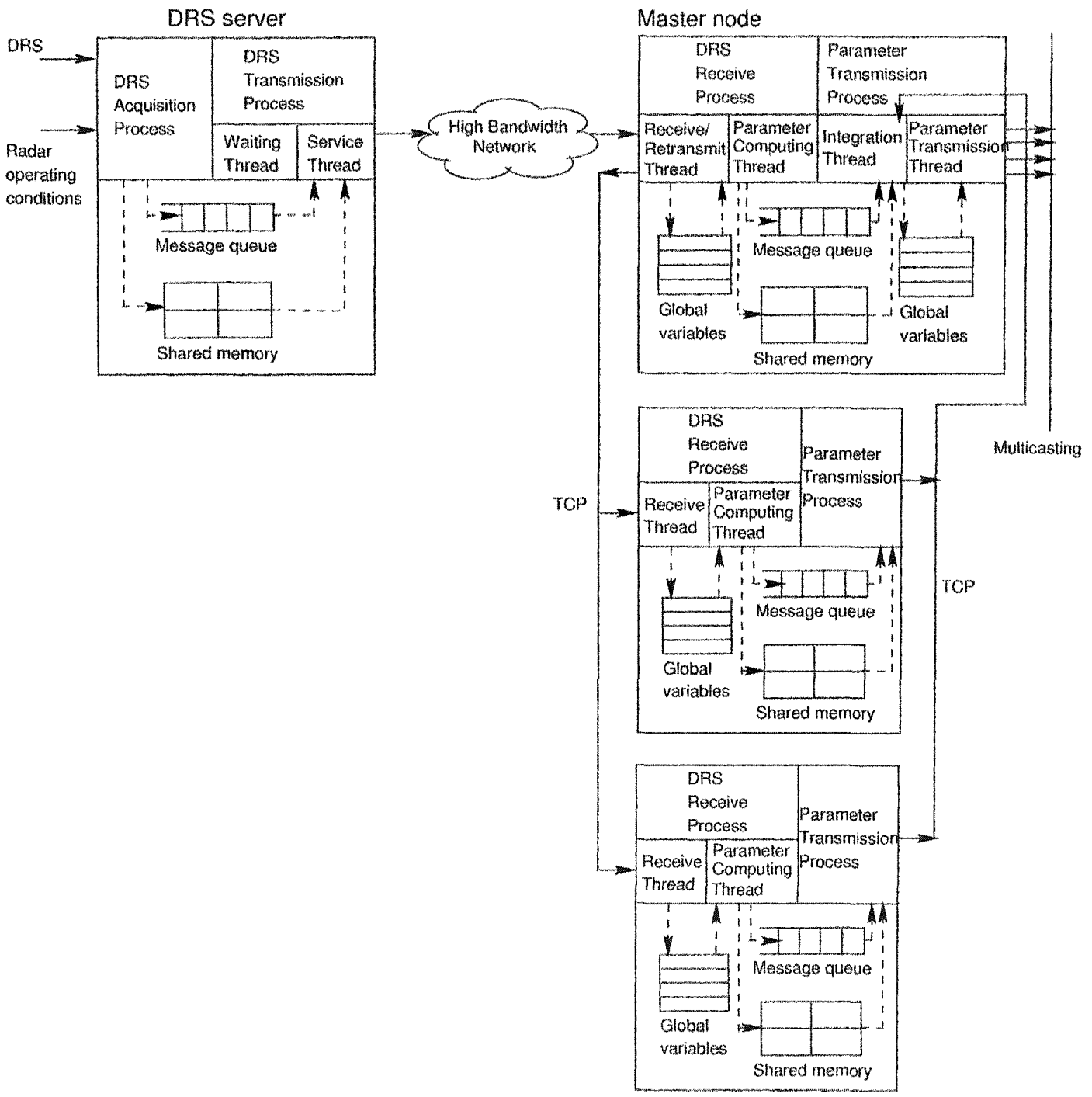


Figure 5.3: Network and end-system architectures for the distributed DRS receiver.

The signal processing and transformations can be implemented within the parameter computing thread. Other TCP connections are also established between the integration thread in the master node and the parameter transfer process in the distributed nodes. The other distributed nodes transmit the estimated radar parameters to the integration thread in the master node. The integration thread in the master node assembles the parameters of all gates and delivers them to the display nodes with the multicasting protocol. To establish multiple TCP connections, the integration thread creates a TCP server socket even though the other distributed nodes transmit the parameters to it.

5.2.2 Communication among the distributed nodes

When the master node distributes the partial DRS to the other distributed nodes, it utilizes the same ray header format as that described in Table 2.4. The master node changes only the values of 'Ngates' and 'Ndata set' fields before re-transmitting the ray header to other distributed nodes. The former contains the number of gates and the latter contains the first gate number included in the user payload, respectively. Usage of the latter field is temporarily changed for the communications among the distributed nodes, which is necessary when the partial parameters are integrated in the master node. The distributed nodes can recognize the amount of data transferred and the first gate number after reading the received ray header. Table 5.1 provides the header format for the transmission of the estimated parameters to the master node. Depending on the operation mode, the data block structures sent to the master node have different parameters as shown in Fig. 5.4.

The integration thread in the master node recognizes the parameters of an identical ray by comparing the field of 'Start angle' field in the header after receiving the partial parameter. The integration thread assembles complete set of

Node ID	Integer, Sequentially assigned to the distributed nodes
Start angle	Unsigned Integer
OP mode	Integer, 0 - V only, 1 - H only, 2 - VH, 3 - VHS
Number of gates	Integer
Start gate number	Integer
End gate number	Integer

Table 5.1: Header structure sent by other distributed nodes for the transmission of the estimated parameters.

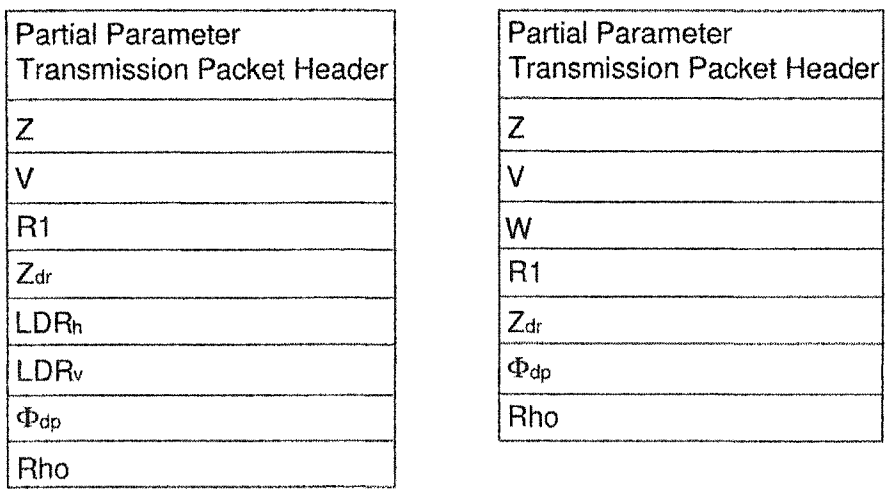


Figure 5.4: User payloads for the transmission of radar parameters to the master node. The header occupies 24 bytes. However, the amount of the parameter data changes depending on the number of gates and the operating mode.

parameters of all gates by looking at the fields of the Node ID, the Number of gates, and the start gate number. The assembled radar parameters are delivered to the display nodes over multicasting protocol.

5.3 Implementation and performance evaluation

The increase in computing capability of the DRS client was demonstrated by evaluating the end-to-end TCP throughput between the DRS server and the master node of the DRS client. Figure 5.5(a) illustrates a simple test bed consisting of end systems connected to each other over Gigabit Ethernet. The end system is a standard PC containing Intel Xeon dual Processors (2.2 GHz). Operating system is Red Hat 8.0 Linux. Intel PRO/1000 Gigabit network interface card (NIC) is plugged into a 64-bit/66-MHz PCI slot in the PC.

Figure 5.6 shows that the peak throughput between the Linux boxes shown in Fig. 5.5(a) is approximately 940 Mbps. The DRS server program runs on the Linux box 1, and the DRS client was made up of either one or two Linux boxes. Table 5.2 shows the end-to-end TCP throughputs between the DRS server and the master node by changing the configuration of the distributed DRS client. The end-to-end TCP throughput between the two Linux boxes was 958.4 Mbps without estimating parameters in the DRS client, which is almost the maximum throughput achievable with the Gigabit NIC. The little lower performance of the Netperf than that of the measurement might come from the additional cycles consumed for the evaluation of the performance in Netperf. On the other hand, when the parameters are estimated in a single node of DRS client, the throughput was reduced to 348.5 Mbps, which shows the same result with that obtained via emulation as described in Section 3.5. When the DRS client consists of two Linux boxes, the throughput was increased up to 766.7 Mbps. This result clearly shows that when the high-bandwidth VCHILL applications fail to operate in real-time

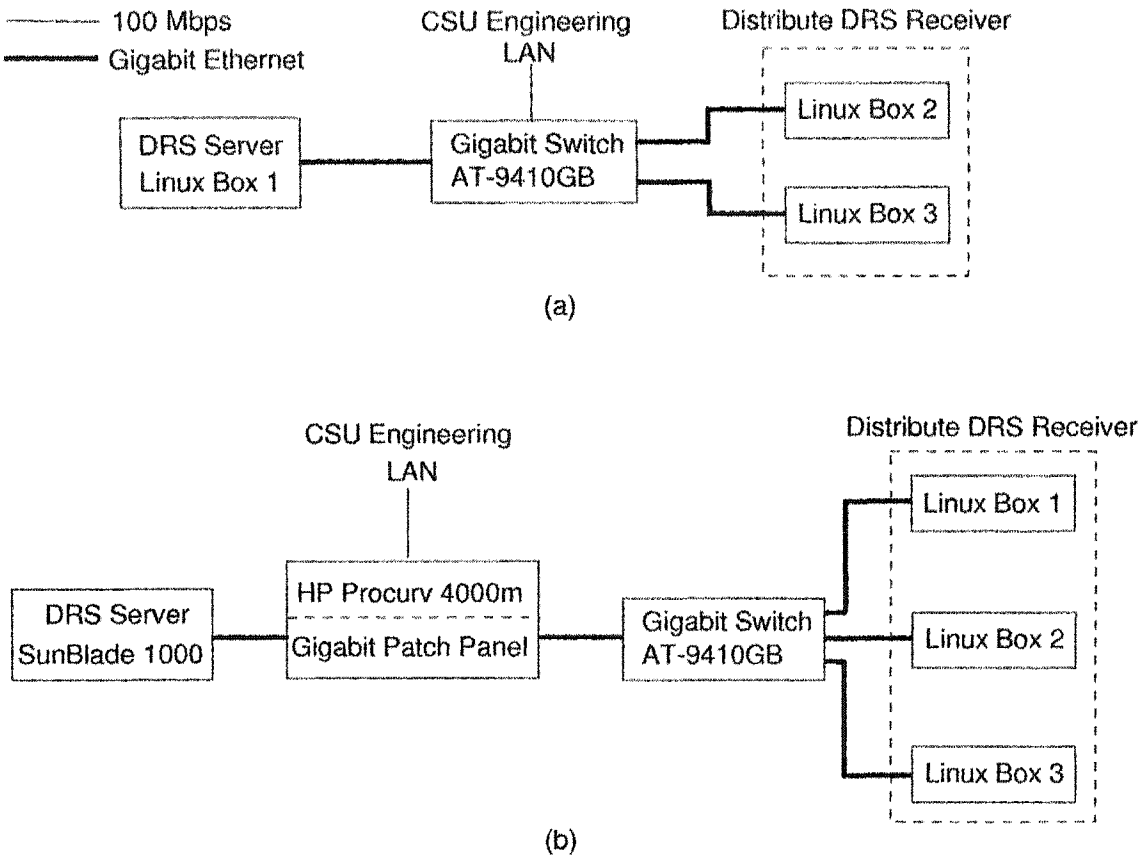


Figure 5.5: Test bed for evaluating performance of the distributed DRS client. The end systems are connected to each other through Gigabit Ethernet.

because of the insufficient computing capability of a DRS client, the distributed DRS client can be a promising solution.

Here, it should be noted that the throughput between the DRS server and client should not be a bottleneck to obtain the expected results. Figure 5.5(b) shows another test bed consisting of end systems and two Gigabit Ethernet switches. The DRS server is a Sun workstation (SunBlade 1000) and the distributed DRS client is comprised of three Linux boxes that were utilized in the previous evaluation. The Sun workstation has dual UltraSPARC-III processors (750 MHz) running with Solaris 8 operating system. A Gigabit Ethernet NIC (SK-9821) from SysKonnect GmbH is plugged into a 64-bit/66-MHz PCI slot. The maximum available

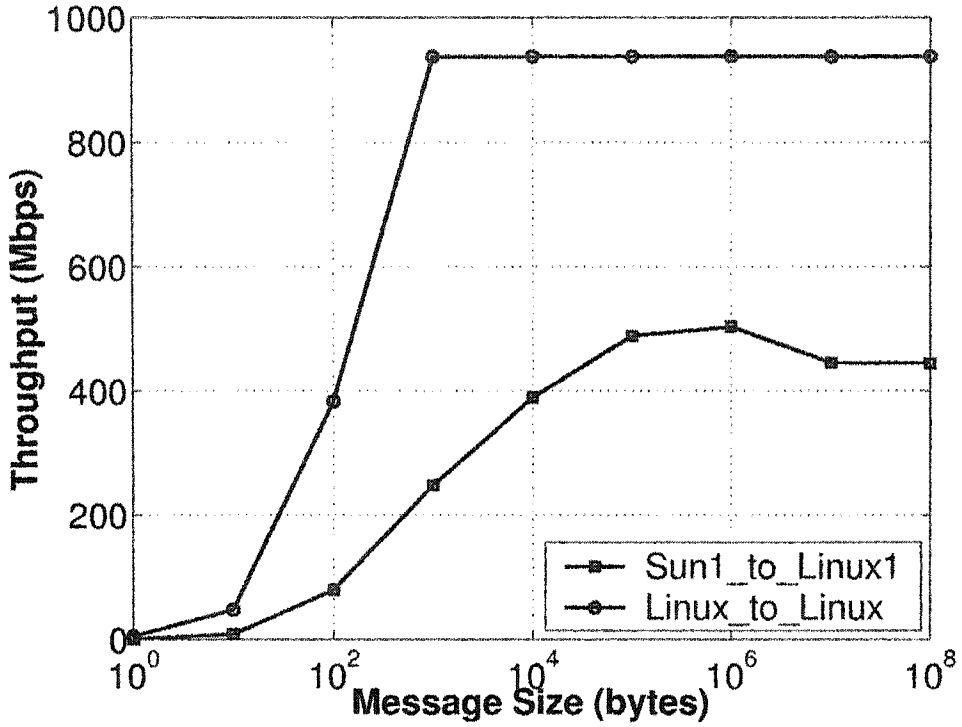


Figure 5.6: Performance for the Gigabit links.

Configuration	Mean End-to-End TCP throughput (Mbps)
Linux Box 1 → Single client Linux Box 2 (No computation)	958.4
Linux Box 1 → Single client Linux Box 2	348.5
Linux Box 1 → Distributed client with two Linux Boxes	766.7

Table 5.2: End-to-end TCP throughput varying the configuration of the DRS client when the link capacity between the DRS server and client is not limited.

Configuration	Mean End-to-End TCP throughput (Mbps)
Sun W.S. → Single client Linux Box 1 (No computation)	308.1
Sun W.S. → Single client Linux Box 1	314.5
Sun W.S. → Distributed client with two Linux Boxes	316.7
Sun W.S. → Distributed client with three Linux Boxes	319.0

Table 5.3: End-to-end TCP throughput varying the configuration of the DRS client when the link capacity between the DRS server and client is limited.

throughput between the Sun workstation and the master node with the block size of 8000 bytes was approximately 350 Mbps as shown in Fig. 5.6.

End-to-end TCP throughputs between the DRS server and the master node were also measured by changing the DRS client configuration. Table 5.3 shows the DRS client configurations and associated end-to-end throughputs. The end-to-end TCP throughput between the Sun Workstation and the Linux Box 1 was 308.1 Mbps when the client did not compute the radar parameters. When a single node of DRS client computed the radar parameters, the throughput was 314.5 Mbps. The result seems to be contrary to the previous trends in which the end-to-end throughput was decreased because of the computation. Here, the throughput was even slightly increased by the computation. This might be caused by the effect of the hyper-threading as introduced in Section 2.5. When the workload is light, most of the CPU capability is idle and the entire performance of the system is low. However, when the workload is increased, the hyper-threading function starts to operate and the entire performance increases.

Another interesting result is that the operation of distributed DRS client did not increase the throughput significantly. The throughput was only 319 Mbps when three Linux boxes collaborated as a DRS client. This result is caused by the low maximum throughput achievable between the DRS server and the master

node. As shown in Fig. 5.6, the maximum throughput between the Sun workstation and the Linux box is approximately 350 Mbps when the data size is 8000 bytes. The performance of the copper-based Gigabit Ethernet depends on PCI bus speed, NIC, variations of drivers and of operating systems, and size of MTU [47]. This low maximum throughput might be caused by the link passing through the Gigabit patch panel, heterogeneity of the end systems, and necessity of byte-order conversion. Here, the low performance cannot support enough transmission rate to obtain the anticipated effectiveness of the increased computing capability. Therefore, reserving sufficient link capacity between the DRS server and client is critical to obtain the effectiveness of the distributed DRS client. Note also that optimization of the workload balance will be necessary to maximize the computing performance.

5.4 Summary and Conclusion

Intense signal processing on the received DRS can make the end-to-end TCP throughput reduced because of the insufficient computing capability of an end system, which might eventually result in failure in real-time operation of the high-bandwidth VCHILL. To increase the computing capability, the concept of the distributed DRS client was developed. The essential idea of the distributed DRS client is to integrate individual computing capability via internetworking. The design for the distributed DRS client, which includes the end-system architectures and the user payload structures for the communication among the distributed nodes, was developd relying on the operation principle of the pulsed-Doppler radar and the theory of radar parameter estimation. A master node evenly distributes the computation workloads to other distributed nodes, and integrates the estimated parameters of partial gates into complete sets including all gates. The end-to-end performances were evaluated over test beds having different network and end-node configurations. The increase in the computing capability has been clearly

demonstrated by achieving approximately two times higher end-to-end throughput with the distributed DRS client consisting of two Linux boxes than that of a single node of DRS client. Reserving high peak throughput between the DRS server and the master node among the DRS client is critical to obtain the effectiveness of the distributed client system. For the future applications of the high-bandwidth VCHILL which require intense computing capability, the concept of the distributed DRS client can provide a promising solution to insufficient computing capability of an end-system.

Chapter 6

STUDY ON THE NETWORKED RADAR SYSTEM

6.1 Introduction

Spatially distributed radar systems can be used for either observation of same targets with multiple radars, such as concurrent multiple Doppler observations, or replacement of a large centralized radar with a cluster of small radars. Multiple Doppler radars provide very useful information about targets, such as multidimensional motion of the scatterers [8, 2]. Benefits obtained by operating clustered radar systems compared with the conventional single radar operation have been also pointed out in terms of increase in reliability of measurements and information quality. As an earlier study introduced, the distributed radar system can also provide benefits in radar system design, such as great reductions in cost brought about by low-power transmitters, cheaper receivers, as well as smaller antennas, radome and pedestal size [6].

These distributed radars must be deployed in a networked environment so that data exchange between the nodes within a specified time should be feasible. Advances in networking technologies and explosive increases in computing capability, which have revolutionized the prevailed paradigms in various fields, might enable the notion of the distributed radar operation. However, implementation of the networked radar system involves challenges, such as multiple streams of information at high bit rates, strict real-time requirements for data transmission and computation, need for spatial and temporal synchronization of information distribution,

and the variable latencies introduced by physical distances, router delays and end-system capabilities [6]. Preliminary studies on the networked radar operation are shown in [7, 48].

In this chapter, we present operational models, end-system architectures, and network models to make operation of the networked weather radar system feasible providing end users with real-time data. The operational models explain how to operate and manage the networked radar systems in order to obtain their benefits. The end-system architectures associated with the operational models are developed relying on the operation of multi-processes and multi-threads, which have been enabled by the advances in computing capability and operating system. Various options for the network models are available. However, the wireless link-based network models, such as wireless LAN bridge and satellite link, will be mainly examined considering mobile platformed radar systems. Recently, it was announced that the wireless technology enables long distance data transmission with high data rate [49, 50]. Moreover, satellite communication systems can achieve uplink transmission capacity up to 2 Mbps [51, 52, 53]. The evaluation of the factors affecting the performance will give an insight into critical issues for implementation. In addition, the scalability for networked radar system will be examined in terms of power consumption for pulse transmission, computation workload, and communication workload varying Quality of Service (QoS) parameters and maximum operation range of individual radar. These scalability metrics are defined in combination with radar operating parameters and ratio of covering radius of a large radar to that of a networked radar. Finally, some challenging issues to design the networked radar system and develop application layer protocols to provide the high QoS real-time radar data will be discussed.

6.2 Operational models

The operational models are developed based on the following assumptions. Radar parameters are estimated at the radar sites and transferred to the integration center. Coherency across the radars is maintained by locking all the stable local oscillators of the radars to a common GPS clock. A control center and an integration center operate at the same physical location, which can be either one of radar sites or other places. The control center computes scanning parameters for optimized radar operation, and the integration center combines radar parameters delivered from radar sites. Cartesian coordinate system is chosen and all radar data are converted from the natural polar coordinates to Cartesian before manipulating data.

6.2.1 Case A: Observation of common volume

The multiple radar observation measuring the multi-dimensional Doppler velocity is an example of this application. When two radars are observing the same volume, the common volume where the angle between the beams from the two radars exceeds 30^o, are referred to dual Doppler lobes as shown in Fig. 6.1 [2]. For a given observation volume, the scan-optimization procedure specifies which radars will be involved in measuring and optimal radar scan parameters are accordingly decided. The optimization depends on the temporal and spatial scale of the event as well as scan rate [54]. The optimized radar scan parameters are delivered to the radar control systems to initiate the observations.

The estimated radar parameters are temporarily stored in buffer at each radar site. Each radar site sends the coordinate information associated with the measurement just after completing a volume scan to the integration center. The integration center converts the local radar polar coordinates to the common Cartesian coordinate. Then, it compares the coordinates to find out the overlapped regions.

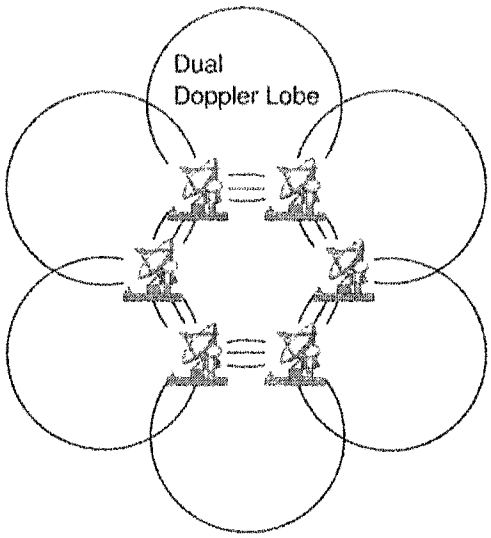


Figure 6.1: An example of distributed radar locations for observing same targets.

The existence of the overlapped coordinates means the availability of data on the regions observed by more than one radar. Then, it sends the information of the overlapped coordinates to the radar sites. The radar sites transmit only the radar parameters associated with the overlapped regions to the integration center. The integration center creates the multi-dimensional information by integrating the received data. The unit processing time of this operation is the scan time of the volume. The development of a fast algorithm to integrate the parameters would be critical to ensure acceptable performance.

6.2.2 Case B: Replacement of a large radar

Figure 6.2 illustrates the geometric model of the application replacing a large radar with networked multiple small radars. The configuration of distributed radars is similar to the cell structure model of wireless communications. The number of networked radars is linearly proportional to the ratio of the covering radius of a single radar to that of small radar. The equation for deriving the number of networked radars can be written as,

$$N_{radar} = 6n - 5 \tag{6.1}$$

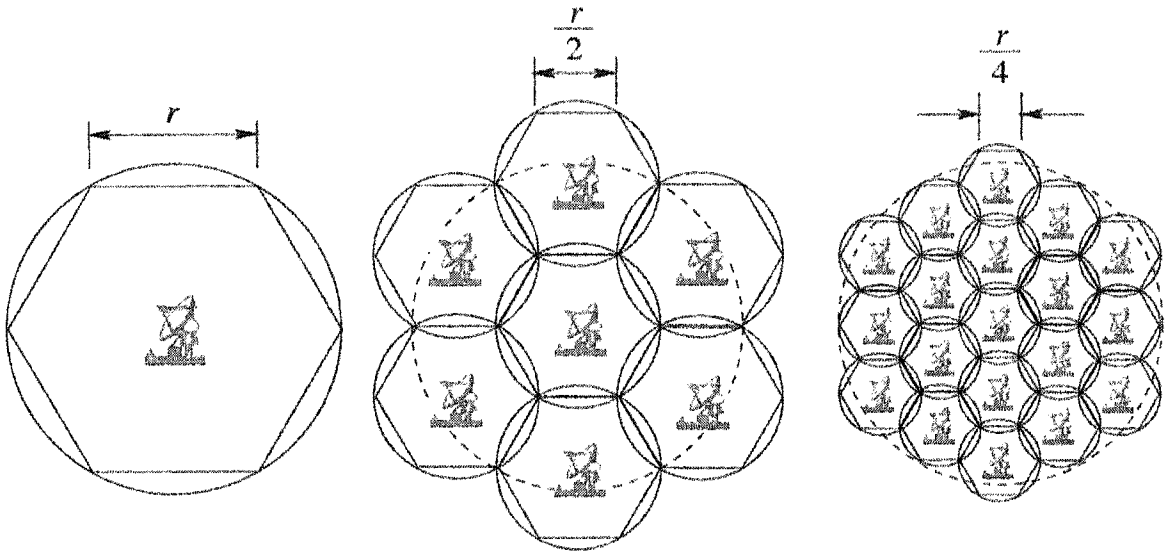


Figure 6.2: Conceptual diagram of the networked radar system. (a) a large radar with the covering radius of r km, (b) a networked radar system consists of 7 radars with the covering radius of $\frac{r}{2}$ km, and (c) a networked radar system consists of 19 radars with the covering radius of $\frac{r}{4}$ km

where n is defined as $n \equiv \frac{R_{max}}{r_{max}}$. Here, R_{max} is the covering radius of the large radar and r_{max} is that of the small radar constructing a clustered radar system. Table 6.1 shows the number of radars necessary to build the networked radar configuration depending on the covering radius of radars.

The control center specifies the scan parameters for the radars based on the user requirements. Radar parameters are estimated with the digitized radar signal at a radar site. The radar parameters are transferred to the integration center accompanied by the coordinates, where the radar parameters are integrated into a complete scan image. The parameter values of the overlapped volumes are averaged. The unit-processing time of this application is the time taken to complete a sweep scan [7].

Radius ratio $\frac{R_{max}}{r_{max}}$	Number of radars N_{radar}
1	1
2	7
4	19
8	43

Table 6.1: Number of radars required to replace a large radar with networked small radars varying the covering radius of the small radar.

6.3 End-system architectures

At the radar site, multiple processes, namely data acquisition, computation, and transmission processes run simultaneously as shown in Fig. 6.3(a). The data acquisition process receives the sampled data and the radar operating conditions from the digitizer and the radar controller, respectively. The computation process estimates the radar parameters from the digitized radar signal. The transmission process maintains a connection to the integration center and sends the radar parameters referring to the coordinates. Data exchanges between the processes are performed through the shared memories. Depending on the computing capability, the data acquisition and other processes can operate on distributed computing resources as described in [9].

In the case of A, receive/transmission, integration, and transmission processes run simultaneously on the integration center as shown in Fig. 6.3(b). The coordinates conversion/comparison process converts the polar coordinate to the common Cartesian coordinate and find out the overlapped regions. The overlapped coordinate information are delivered to both the radar sites and the integration process. The receive/transmission process is responsible for communicating to the radar sites. The integration process computes the multi-dimensional data sets and passes them to the transmission process that is connected to other data networks,

such as the Internet. The multi-dimensional radar parameters are distributed to remotely located nodes for display or other application programs by the transmission process.

In the case of B, receive, coordinates conversion/integration, and transmission processes operate simultaneously on the integration center as shown in Fig. 6.3(c). The receive process is responsible for receiving radar signal parameters from the individual radar site. A ray data set is moved to the coordinates conversion/integration process upon the arrival of the data set at the receive process. In this application, the complete scan image is integrated over a virtual radar sweep and the integrated parameters are delivered to the remotely located nodes for display or other application programs by the transmission process.

Figure 6.4 shows the timing sequence of the processes in the integration center. One sweep time is equivalent to a unit-processing time at which all processes synchronize. Integration procedure starts upon the receipt of the first parameter data from any radar site and is completed when the last data arrives at the integration center. The time lag between the receiving and integration can be ignored compared with the unit processing time. The transmission process starts sending the integrated radar image data upon the completion of integration, which introduces a time lag of one sweep time between the actual radar operation and the display. While the display is conducted ray by ray for the single radar operation, the display mode can be flexible for the networked radar operation according to the design. However, the time taken for scanning a sweep should be equal if the sweep speeds are identical for both the single radar and the networked radar operation. On the assumption that the sweep speed is 6 degrees/sec, the unit-processing time is 60 sec.

6.4 Network model

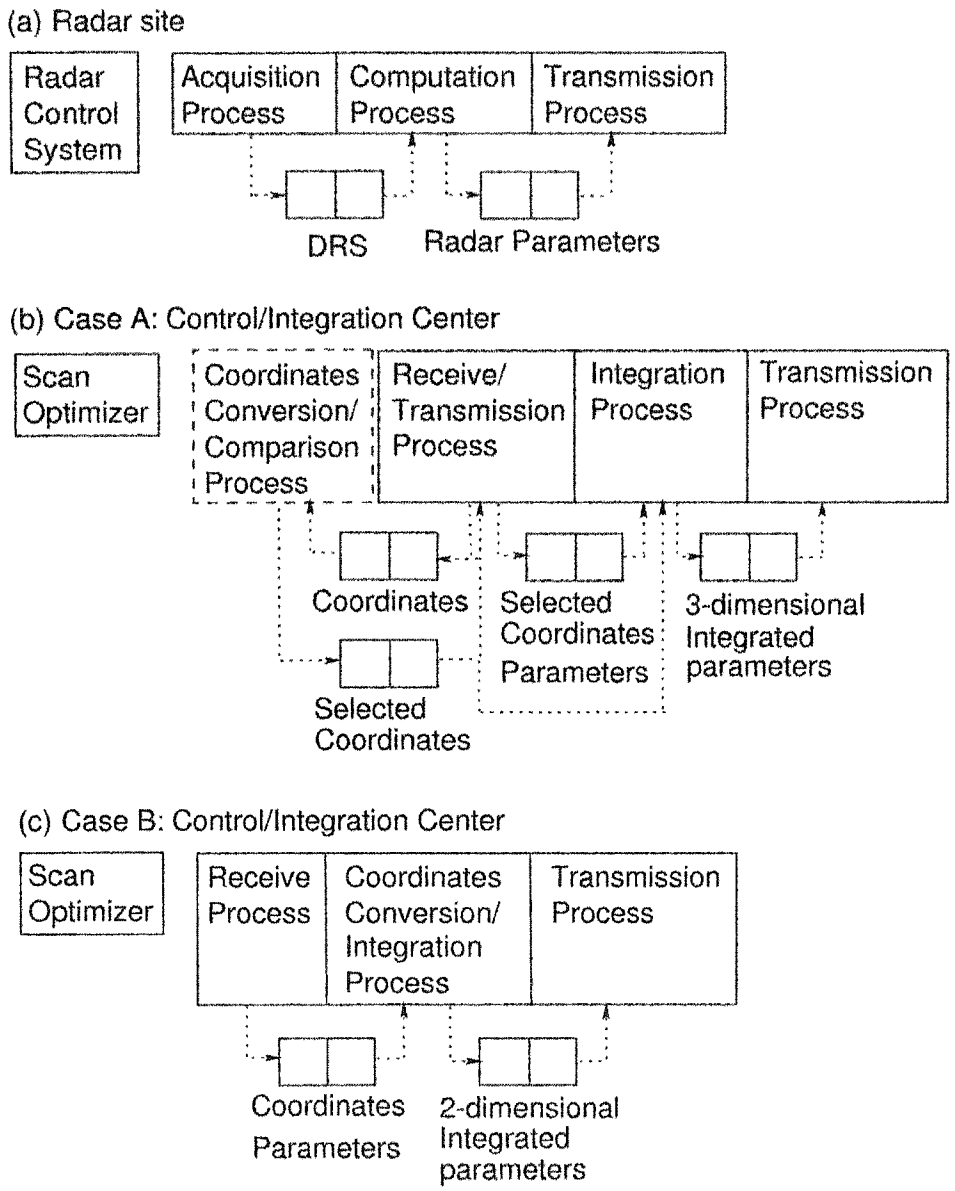


Figure 6.3: End-system architecture. Data are exchanged between the processes through shared memory.

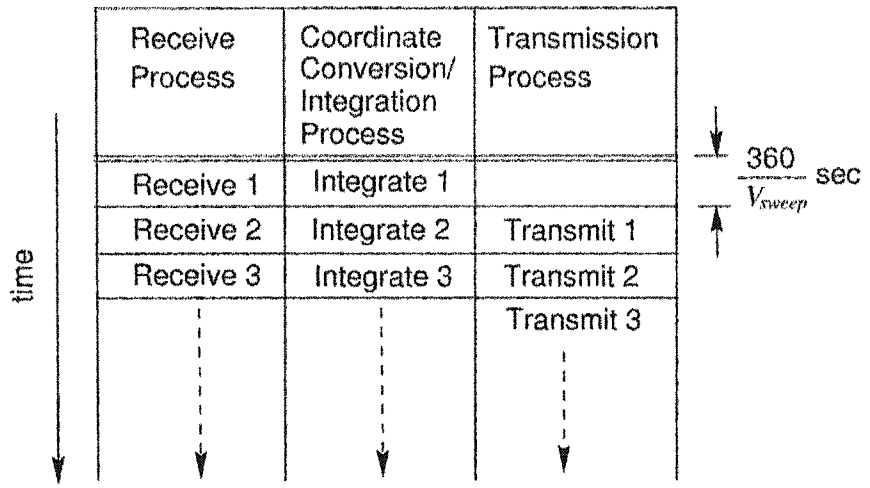


Figure 6.4: Timing sequence of the three processes operating in the integration center simultaneously. V_{sweep} stands for the sweep speed in degree/sec.

6.4.1 Data rate of networked radar system

Before considering, it is informative to estimate the required bandwidth to exchange the data between the networked radar sites and the control/integration center. Based on the operational model of the case B, the bandwidth to meet the requirement for real-time operation can be estimated.

The amount of data generated in each radar site are calculated as shown in Eq. (6.2).

$$\begin{aligned} D_{sweep} &= D_{ray}(\frac{360}{\Delta\theta}) \\ &= (D_{header} + N_{gate} \cdot N_{parameter})(\frac{360}{\Delta\theta}), \end{aligned} \tag{6.2}$$

where D_{sweep} is the data amount of a sweep; D_{ray} is the data amount of a ray; D_{header} is the size of ray header; N_{gate} is the number of gates; $N_{parameter}$ is the number of radar parameters; and $\Delta\theta$ is the angular resolution in degree. The N_{gate} is calculated as shown in Eq. (6.3).

$$N_{gate} = \frac{2R_{sample} \cdot \tau_{max}}{c}, \tag{6.3}$$

where R_{sample} is defined as the sampling rate of a radar; r_{max} is the radius of area under coverage in meters; and c is the speed of light in m/sec.

As an example, we can assume that a large radar with the coverage radius of 150 km is replaced with seven small radars with the the coverage radius of 75 km. The following assumptions on the operating conditions are made. a) the sampling rate is 1 MHz; b) the number of radar parameter is 10; c) the angular resolution is 0.75 degrees; and d) the ray header size is 150 bytes. Then, the number of gates is 500 and the total amount of data for one sweep is 2.472 MB. It is also assumed that the sweep speeds of all networked radars are same (however, not necessary). If the sweep speed is V_{sweep} in degrees/sec, the time taken to display one sweep radar image (one sweep time) is $\frac{360^o}{V_{sweep}}$ sec. When the sweep speed is assumed to be 6 degrees/sec, the time taken to display one sweep radar image is 60 sec. Given the total amount of data for one sweep, the minimum throughput to meet the operational scenario is 329.6 Kbps.

6.4.2 Wireless LAN bridge-based network model

Mobile-platformed radar systems can provide great flexibility and simplicity in configuring physical locations of the networked radars [55]. This mobility can provide a variety of data collection for various experimental purposes. Since the position of the radar can be changed depending on the application, a wireless link is suitable for networking the mobile-platformed radars. In contrast, wired networks demand considerable expenses, not only for constructing initial network infrastructures, but also for maintenance.

The commercial wireless vendor market is rapidly advancing and wireless Ethernet bridges are able to achieve data rates higher than 10 Mbps and long data transmission ranges longer than 40 miles [49]. In addition, a variety of products based on the wireless LAN technologies, in which the data rate ranges from 1

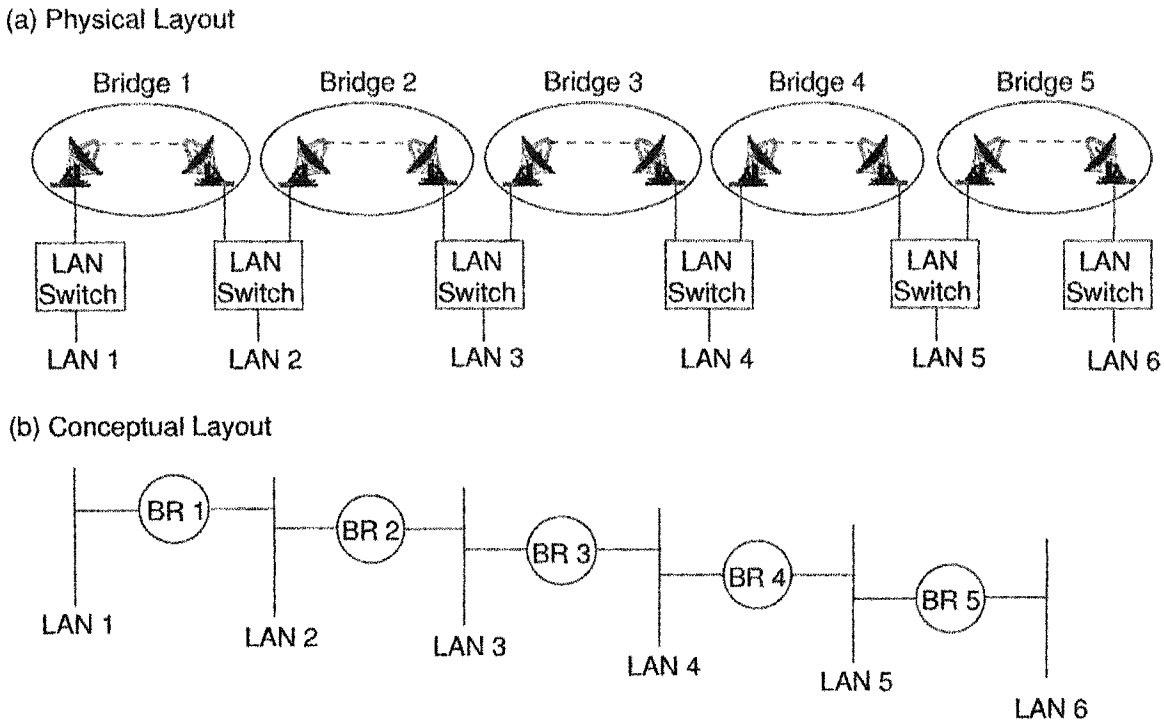


Figure 6.5: A wireless LAN-based network model for the operation of distributed radar system.

Mbps to 54 Mbps and outdoor range is from 5 km to 40 km, is currently available in the commercial market [50]. Capabilities and performance of the wireless communication systems are expected to be improved continuously.

Figure 6.5 illustrates a wireless LAN bridge-based network design for the deployment of distributed radars. The essential function of the wireless LAN bridge is to connect sharing LANS over wireless link. It is assumed that individual LAN to which the end systems are attached is available at each radar site. These LANs are connected to each other via the wireless Ethernet bridge, which leads to a networked LAN topology. A gateway to the Internet can be connected to the networked LAN, such that the integrated radar parameters can be delivered remote locations for further applications. This network model can be applied to any multiple radar operations if the line-of-sights are maintained between the communication antennas.

6.4.3 Satellite link-based network model

The satellite communication links can be considered as an alternate network model because of more flexibility in the environment of unpredictable topography where the radar systems are deployed. For the wireless LAN bridge, the line-of-sights must be maintained between two antennas. However, the satellite link can usually establish the communication link without interruption between ground and satellite. Recently, it was announced that some satellite communication systems achieve uplink data rate more than 2 Mbps [51, 52, 53]. The satellite link-based network model for the networked radar operation is illustrated in Fig. 6.6. The estimated radar parameters are transmitted to a satellite from a ground terminal, like Very Small Aperture Terminal (VSAT), located at individual mobile radar site. The satellite, which can perform on-board processing (OBP), retransmits the data to a gateway on the ground through a high-bandwidth downlink. Then, the data is transferred to the integration center through the Internet. The control data, such as optimized parameters for radar scanning and overlapped coordinates, can be delivered in reverse direction.

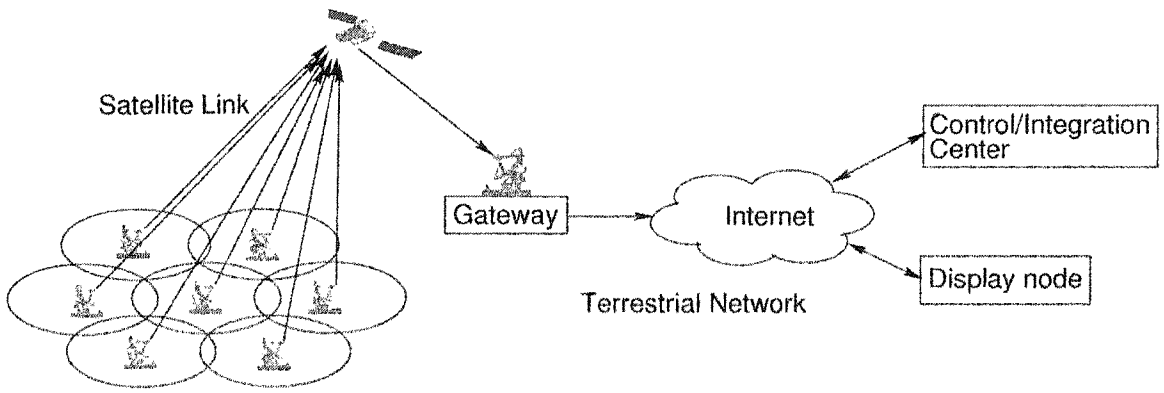


Figure 6.6: Proposed satellite link-based network model for exchanging data between the radar sites and the control/integration center.

Although the satellite communication link has strong features, for instance rapid and flexible network configuration, wide coverage of service area, multiple

Satellite obit	GEO	MEO	LEO
Propagation delay of satellite links (ms)	250	110	20
Propagation delay of terrestrial links (ms)	42	42	42
RTT (ms)	584	304	124
Max. Throughput (Kbps)	270	513	930

Table 6.2: Maximum achievable throughput without any congestion and error with TCP Reno. These data were obtained via 'ns' simulations.

access capability and multiple distribution capability, it has weaknesses as well, namely long propagation delays and error bursts [56]. Therefore, an examination on the validation for the proposed operational model is required. For the transmission over TCP, the end-to-end throughput is determined by the RTT and the network traffic condition. Propagation delays of geostationary earth orbit (GEO), medium earth orbit (MEO) and low earth orbit (LEO) are approximately 250-280 msec, 110-130 msec and 20-25 msec, respectively [57]. Table 6.2 shows the highest end-to-end TCP throughputs achievable depending on the type of satellite link under the assumption of no congestion and packet drops. The end-to-end TCP throughput changes greatly depending on the types of the satellite links because of the difference in RTT. Considering the example mentioned in Section 6.4.1, the required bandwidth is around 330 Kbps. Therefore, transmission over the GEO satellite link demands data compression to meet the requirements for real-time operation because of the low utilization of the available bandwidth caused by the long propagation delay.

As described in Section 6.3, all radar parameters of a sweep at the radar sites should arrive at the integration center during a unit-processing time. In this application, end-to-end packet delay jitter occurred during the unit processing time is not critical. However, it is important that entire sweep data from all networked radar sites should arrive at the integration center by the end of a unit-processing

time. Otherwise, the synchronized processing of the receipt and integration cannot be accomplished. Using a receiving buffer of appropriate size might be able to alleviate the problem, while introducing more time lag between the actual scanning and display.

6.5 Scalability of the networked radar system

6.5.1 QoS Parameters

The important QoS parameters for the weather radar applications are spatial resolution and data accuracy assuming that variance of measurements is ignored. The spatial resolution represents the volume a data point covers as shown in Fig. 6.7. The range resolution is determined by the pulse width that is a radar operating condition. The Azimuthal beamwidth resolution and the elevation beamwidth resolution are determined by both the radar operating conditions and the measuring range. For the networked radar system, the measuring range is confined in the shorter radius in comparison with the single radar operation. This leads to the higher spatial resolution than the single radar operation when the pulse width angular resolution are assumed to be identical for both operation. The angular resolution is calculated from Eq. (6.4).

$$\texttt{Angular resolution} = V_{sweep} \times N_{pulse} \times T_s \tag{6.4}$$

where N_{pulse} is the number of pulses transmitted at a particular angle, and T_s is the PRT. For example, when the sweep speed is 6 degrees/sec, N_{pulse} is 128, and the PRT is 1 msec for the PPI scan, the Azimuthal angular resolution is 0.768 degrees.

The data accuracy, which is evaluated with the variances of the parameters, is determined by the number pulses transmitted at a particular pointing angle. As explained in Section 3.2, the variances of the parameters are inversely proportional to the square of the number of samples that are involved in estimation.

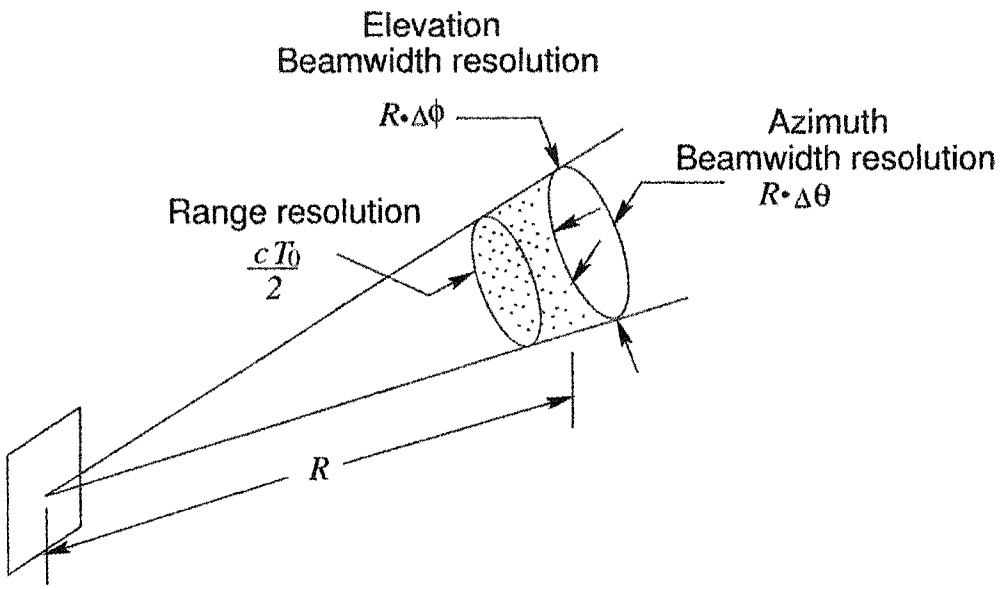


Figure 6.7: Conceptual digram showing the spatial resolution which is represented by range resolution, elevation beamwidth resolution, and Azimuth beamwidth resolution. $\Delta\theta$ and $\Delta\phi$ are the angular resolutions in Azimuth and elevation directions, respectively. T_0 is the pulse width, R is the measuring range, and c is the speed of light.

6.5.2 Scalability changes with radar size ratio

The scalability can be examined in terms of power consumption for radar pulse transmission, computation workload, and communication workload varying the QoS and the number of radars. The power consumption is a metric associated with the peak transmitted power of the radar system. The radar range equation described in Eq. (6.5), which explains the fundamental design concepts of radar, shows that the peak power of radar is proportional to the square of the radius to get the same received power [12].

$$P_r(r_0) = \left[\frac{cT_0}{2}\right]\left[\frac{\lambda^2 P_t G_0^2}{(4\pi)^3}\right]\left[\frac{G_r}{l_r}\right]\left[\frac{\pi\theta_1\phi_1}{8\ln 2}\right]\left[\frac{\eta(\gamma_0)}{r_0^2}\right], \tag{6.5}$$

where P_r is the received signal power; P_t is the transmitted signal power; c is the speed of light; T_0 is the transmit pulse width; G_r is the receiver gain; l_r is the loss due to receiver filter; λ is the wavelength of the radar signal; G_0 is the antenna gain; r_0 is the measuring range; η is the volumetric radar cross section (per unit volume); and θ_1 and ϕ_1 are the half power beamwidth in Azimuth and elevation direction, respectively.

The computation workload is a metric associated with the total amount of computing resource necessary for the estimation of radar parameters. To compute the parameters for a gate, the covariance matrix elements are estimated over the samples of the gate. The amount of the computation is proportional to the product of the number of samples and the number of gates.

The communication workload is a metric associated with the amount of estimated parameters that are transferred to the integration center. As described in Section 6.4, the user payload of the transferred data consists of the display header and the whole parameters that are estimated.

The proposed metrics for the networked radar systems are defined in combinations with radar operating parameters as well as the radar size ratio(n) that is given by Eq. (6.1) as belows.

	Scenario 1	Scenario 2	Scenario 3
Outermost Azimuth Beamwidth Resolution	Same	Increase $\times 2, \times 4, \times 8$	Same
Data Accuracy	Same	Same	Increase $\times \frac{9}{4}$

Table 6.3: Different scenarios for comparison. The outermost Azimuth beamwidth resolution is the resolution at the edge of the region a radar covers.

- Radar power consumption

  $\propto \alpha r_{max}^2 \times N_{radar} \times$ `Pulse number/sweep`

  $\propto \left[\frac{360\alpha}{V_{sweep} \times T_s}\right] R_{max}^2 \left[\frac{6}{n} - \frac{5}{n^2}\right]$

- Computation workload

  $\propto N_{radar} \times N_{gate} \times$ `Pulse number/sweep`

  $\propto \left[\frac{360}{G_{res} \times V_{sweep} \times T_s}\right] R_{max} \left[6 - \frac{5}{n}\right]$

- Communication workload

  $\propto N_{radar} \times (D_{header} + N_{gate} \times N_{parameters})(\frac{360}{\Delta\theta})$

  $\propto (D_{header} + \frac{N_{parameters}}{G_{res}} \times \frac{R_{max}}{n})(\frac{1}{N_{pulse} \times T_s})(6n - 5)$

where G_{res} is the range resolution and α is a constant coefficient.

To examine the scalability, three scenarios varying the QoS specification are considered as shown in Table 6.3. Figure 6.8 shows the changes in scalability metrics normalized decreasing the radius of networked radars. The networked radar system does not increase the metrics compared with the single radar operation when the identical or slightly higher QoS parameters of angular beamwidth resolution and data accuracy are aimed at.

6.6 Discussion

Networked radar systems can experience undesirable noise in received signals due to interference from other nearby radar systems. Phase coding technique can

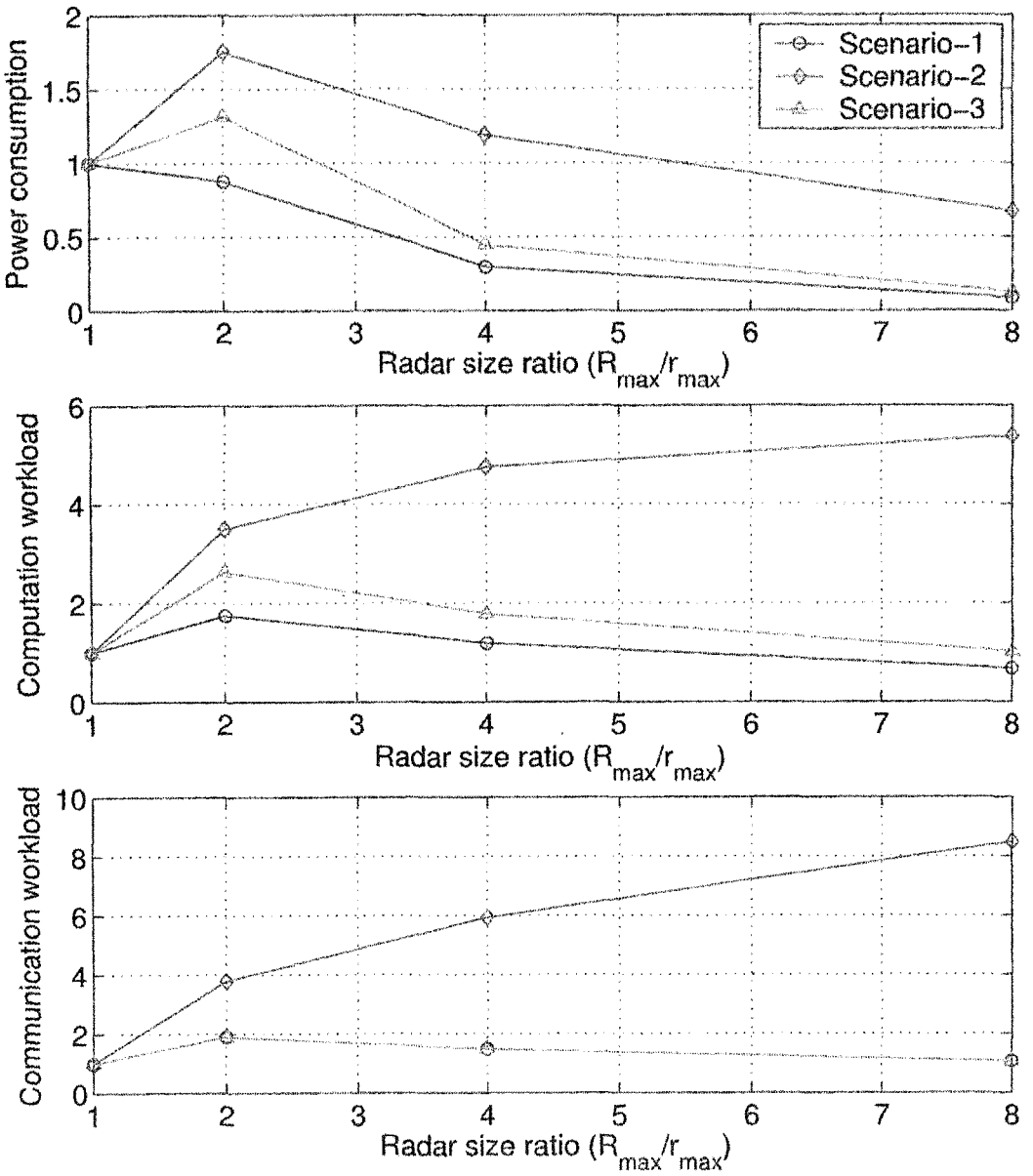


Figure 6.8: (a) Power consumption for pulse emission, (b) Computation workload, and (c) Communication workload ratios with the different scenarios varying number of radars.

be used to remove the effect of the interference. Each radar system selects a specific phase of the emitted radar pulse such that other signals with different phases can be filtered out.

Networked radar system can have various end-user application areas, and end users might want to access the system through the Internet. An application layer management protocol makes the hardware and software of the lower layers transparent to the applications of the networked radar system. The management protocol can include the following administrative tasks, such as introducing the rules related to data aggregation and clustering of the networked radars, exchanging data related to the location finding algorithms, time synchronization of the networked radars, turning networked radars on and off, querying the networked radar system configuration and the status of the radars, reconfiguring the network, authentication, key distribution, and security in data communications [58].

6.7 Summary and Conclusion

The operational models of the networked radar system have been studied for two different applications, namely a) observation of the same volume target by multiple radars and b) the replacement of a large radar with a networked of small radars. Design of the end-system architectures is based on the multiprocesses and multithreads. The end system at the radar site includes the functions of data acquisition, parameter estimation, transmission, as well as radar control. The end system at the control/integration center includes the functions of receive/transmission, coordinate conversion, and integration, and scan optimization. The feasibility of the network models based on wireless LAN bridge and satellite links have been examined to provide a network environment in which the data are exchanged between the nodes within a specified time. Spatial resolution and data accuracy are considered as primary QoS parameters for weather radar applications. Scalability metrics of the networked radar system are not increased

much compared to a single radar operation under the condition of the identical or slightly higher QoS parameters.

Chapter 7

SUMMARY AND CONCLUSION

In this dissertation, the end-system architectures as well as application and transmission protocols for the high-bandwidth VCHILL are developed and experimentally validated with the CSU-CHILL radar. In addition, concepts for the networked radar operation are also studied. Chapter 2 described the digital-IF receiver developed exclusively for high-bandwidth VCHILL. The parallel receiver has been implemented on a standard PC platform with PCI-interfaced high-speed ADC/DDC signal processor, a digital I/O, and a data conversion circuit for AZ/EL. Since the signal processor generates the overall data rate over a few hundreds Mbps in conjunction with the operation of the pulsed-Doppler radar, the integration of the all acquired data into the specified format requires extremely careful timing for synchronized data acquisition and high-bandwidth data manipulation. The parameters obtained from the parallel receiver show good coincidence with those obtained from the DRX receiver as well as linearity of transmission rate proportional to the sampling rate, which demonstrates that the parallel receiver operates properly for the high-bandwidth VCHILL.

Chapter 3 described the design of the end-system architectures for the high-bandwidth VCHILL. The design is based on the typical client/server model, relying on the simultaneous operation of multiprocesses and multithreads to meet the real-time requirements. The data and packet structures are included in the design

as well. The architectures include the functions of the DRS acquisition and the transmission of the real-time DRS over TCP to remote sites on the DRS server, as well as the receive/estimation of the radar parameters and the delivery of the parameters to other display nodes on the DRS client. The application and network protocols implemented here support the real-time operation of the CSU-CHILL radar up to the data rate of 300 Mbps, which is equivalent to the sampling rate of 5 MHz. Comparison of the displays between the high-bandwidth VCHILL system and the DRX system shows good coincidence.

In Chapter 4, a transmission waveform design schemes tailored for the high-bandwidth VCHILL over UDP have been proposed to provide tolerance of the real-time operation and high quality of end products. The waveform design schemes rely on the operating principle of the pulsed-Doppler radar and the estimation theory of the polarimetric radar parameters. The DRS server groups two or three consecutive range sample data sets depending on the radar operation mode, transmits some of the groups, and suppresses other transmissions according to the available bandwidth. Simulation study presents that the proposed data selection scheme is extremely effective in providing high quality end products. The waveform design schemes have been combined with the source-based rate control with a AIPD based on the feedback to develop congestion control algorithm over unpredictable network. The performance of the proposed congestion control algorithm was evaluated via real-time operation of CSU-CHILL radar. The results clearly show that the server dynamically adapts the transmission rate to the available bandwidth. The end-user displays also provide much higher quality in comparison with the case with no congestion control.

In Chapter 5, to increase the computing capability, the concept of the distributed DRS client was developed. The essential idea of the distributed DRS client is to integrate individual computing capability via internetworking. The

design for the distributed DRS client, which includes the end-system architecture and the packet structure for the data exchange and the communication among the distributed nodes, relies on the operation principle of the pulsed-Doppler radar and theory of radar parameter estimation. The increase in the computing capability has been clearly demonstrated by achieving approximately two times higher end-to-end throughput with the distributed DRS client configured with two nodes than that of a single DRS client. High link capacity between the DRS server and the master node among the DRS client is critical to obtain the effectiveness of the distributed client system. For the future applications of the high-bandwidth VCHILL which require intense computing capability, the distributed DRS client can provide a solution to insufficient computing capability of an end-system.

In Chapter 6, the operational models of the networked radar system have been studied for two different applications, namely a) observation of the same volume target by multiple radars and b) the replacement of a large radar with a networked of small radars. Design of the end-system architectures is based on the multiprocesses and multithreads. The end system at the radar site includes the functions of data acquisition, parameter estimation, transmission, as well as radar control. The end system at the control/integration center includes the functions of receive/transmission, coordinate conversion, and integration, and scan optimization. The network models based on wireless LAN bridge and satellite links have been examined to provide a network environment in which the data are exchanged between the nodes within a specified time. Spatial resolution and data accuracy are considered as primary QoS parameters for weather radar applications. Scalability metrics, such as power consumption for pulse emission, computing load, and communication load, of the networked radar system are not increased much assuming the identical or slightly higher QoS parameters.

Bibliography

[1] D. Brunkow. (2003) CSU-CHILL National Radar Facility. [Online]. Available: http://chill.colostate.edu/

[2] D. Brunkow, V. N. Bringi, P. C. Kennedy, S. A. Rutledge, V. Chandrasekar, E. A. Mueller, and R. K. Bowie, "A description of the CSU-CHILL National Radar Facility," *Journal of Atmospheric and Oceanic Technology*, vol. 17, pp. 1596–1608, 2000.

[3] V. Chandrasekar, D. Brunkow, and A. P. Jayasumana, "CSU-CHILL operation over the Internet: Virtual CSU-CHILL," in *The 30th International Conference on Radar Meteorology*, 2001, pp. 58–60.

[4] J. J. Jishnu, "Client-server design to access radar data," Master's thesis, Colorado State University, Fort Collins, Colorado, Spring 2000.

[5] A. Kumar, "TCP, UDP and IP multicasting based applications for real-time access of CHILL radar data - implementations and performance evaluation," Master's thesis, Colorado State University, Fort Collins, Colorado, Spring 2001.

[6] V. Chandrasekar and A. P. Jayasumana, "Radar design and management in a networked environment," in *Proceedings of SPIE*, vol. 4527, 2001, pp. 142–147.

[7] Y.-G. Cho, V. Chandrasekar, and D. A. Vivanco, "Network model for clustered radar operating application," in *Proceedings of Local Computer Networks 2002*, 2002, pp. 72–74.

[8] P. S. Ray, K. K. Wagner, K. W. Johnson, J. J. Stephens, W. C. Bumgarner, and E. A. Mueller, "Tripple-Doppler observations of a convection storm," *Journal of Applied Meteorology*, vol. 17, no. 8, pp. 1201–1212, August 1978.

[9] Y.-G. Cho, V. Chandrasekar, A. P. Jayasumana, and D. Brunkow, "Architecture and implementation for high bandwidth real-time radar signal transmission and computing application," in *Proceedings of SPIE*, vol. 4863, 2002, pp. 87–98.

[10] S. M. Torress and D. S. Zrnic, "Optimum processing in range to improve estimates of Doppler and polarimetric variables on weather radars," in *Proceedings of the 30th International Conference on Radar Meteorology*, 2001, pp. 325–327.

[11] D. A. Brunkow, "A new receiver and signal processor for the CSU-CHILL radar," in *The 29th International Conference on Radar Meteorology*, 1999, pp. 256–258.

[12] V. N. Bringi and V. Chandrasekar, *Polarimetric Doppler Weather Radar: Principles and Applications*. Cambridge University Press, 2001.

[13] Y. Wu and J. Li, "The design of digital radar receivers," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 1, pp. 35–41, January 1998.

[14] "ICS-554 Operating Manual," Interactive Circuits and Systems Ltd., Ontario, Canada, August 2002.

[15] "GC4016 Multi-Standard Quad DDC Chip Data Sheet," GRAYCHIP, INC., August 2001.

[16] A. Rubini and J. Corbet, *Linux Device Drivers*, 2nd ed. O'Reilly and Associates, Inc., 2001.

[17] "PCI-7224/7248/7296 24/48/96-CH Digital I/O Card User's Guide," ADLINK, October 2000.

[18] "ICS-554 Linux Software Development Kit User's Manual," Interactive Circuits and Systems Ltd., Ontario, Canada, August 2002.

[19] "ADLINK PCI/CPCI DAQ Cards Software Installation Guide," ADLINK, October 2000.

[20] Hyper-threading technology. [Online]. Available: http://www.intel.com/technology/hyperthread/

[21] S. C. Viswanatha, A. P. Jayasumana, V. Chandrasekar, and D. Brunkow, "Network transfer, digital signal processing and display of CSU CHILL digitized radar signals," in *Proceedings of SPIE*, vol. 4527, 2001, pp. 125–133.

[22] W. R. Stevens, *UNIX Network Programming Vol. 2: Interprocess Communications*, 2nd ed. Prentice-Hall, Inc., 1999.

[23] ——, *UNIX Network Programming Vol. 1: Networking APIs: Sockets and XTI*, 2nd ed. Prentice-Hall, Inc., 1998.

[24] S. C. Viswanatha, "Network transfer, digital signal processing and display of CSU CHILL radar time series data," Master's thesis, Colorado State University, Fort Collins, Colorado, Spring 2001.

[25] [Online]. Available: http://www.netperf.org/netperf/NetperfPage.html

[26] "Standard performance evaluation corporation." [Online]. Available: http://www.specbench.org/

[27] [Online]. Available: http://www.gnu.org/manual/gprof-2.9.1/html_mono/gprof.html

[28] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-Friendly congestion control," *IEEE Network*, pp. 28–37, May/June 2001.

[29] T. V. Lakshman and U. Madhow, "The Peformance of TCP/IP for networkd with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, June 1997.

[30] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–145, April 2000.

[31] C. Barakat, "TCP/IP Modeling and validation," *IEEE Network*, pp. 38–47, May/June 2001.

[32] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, August 1999.

[33] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "A transport protocol for real-time application," Network Working Group," RFC 1889, 1996.

[34] J. F. Kurose and K. W. Ross, *Computer Networking: A top-down approach featuring the Internet*, 2nd ed. Addison Wesley, 2003.

[35] S. S. Kaiya, "Radar data transfer protocol of CSU-CHILL digitized radar signals," Master's thesis, Colorado State University, Fort Collins, Colorado, Summer 2002.

[36] S. L. Bangolae, "A TCP-Friendly congestion control mechanism for a high-bandwidth radar application," Master's thesis, Colorado State University, Fort Collins, Colorado, Fall 2003.

[37] Y.-G. Cho and V. Chandrasekar, "Real-time weather radar signal transmission," in *Proceedings of IGARSS '03*, 2003.

[38] ——, "Distribution of real-time weather radar signal over high bandwidth network," in *Proceedings of the 31st Conference on Radar Meteorology*, vol. 2, 2003, pp. 911–914.

[39] R. J. Doviak and D. S. Zrnic, *Doppler radar and weather observations*, 2nd ed. Academic Press, 1993.

[40] V. Chandrasekar, V. N. Bringi, and P. J. Brockwell, "Statistical properties of dual-polarized radar signals," in *The 23th International Conference on Radar Meteorology*, 1986, pp. 193–196.

[41] J. W. Green, "Simulation of multivariate radar signals," Master's thesis, University of Alabama in Huntsville, Huntsville, Alabama, Fall 1990.

[42] A. S. Mudukutore, V. Chandrasekar, and R. J. Keeler, "Pulse compression for weather radars," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 125–142, January 1998.

[43] K.-W. Lee, T.-E. Kim, and V. Bharghavan, "A comparison of end-to-end congestion control algorithms: The case of AIMD and AIPD," in *INFOCOM 2001*, 2001.

[44] Nist net home page. [Online]. Available: http://snad.ncsl.nist.gov/itg/nistnet/

[45] [Online]. Available: http://www.myri.com/

[46] [Online]. Available: http://www.mpi-forum.org/

[47] P. Gray and A. Betz, "Performance evaluation of copper-based Gigabit Ethernet Interface," in *Proceeding of the 27th Conference on Local Computer Networks*, 2002, pp. 679–690.

[48] Y.-G. Cho and V. Chandrasekar, "Distributed weather radar system operation over wireless network," in *Proceedings of IGARSS '03*, 2003.

[49] [Online]. Available: http://www.lastmile2.com

[50] [Online]. Available: http://www.simplywireless.com.au/longrange.htm

[51] C. Metz, "Ip-over-Satellite: Internet connectivity blasts off," *IEEE Internet Computing*, pp. 84–89, July/August 2000.

[52] [Online]. Available: http://www.astra.lu/

[53] [Online]. Available: http://www.gilat.com/gilat/

[54] C. D. Burghart, P. J. Wyngaard, P. H. Herzegh, and J. W. Wilson, "A program for optimization of meteorological radar scanning," in *The 24th International Conference on Radar Meteorology*, 1989, pp. 573–576.

[55] K. Iwanami, R. Misumi, M. Maki, T. Wakayama, K. Hata, and S. Watanabe, "Development of a multiparameter radar system on mobile platform," in *The 30th International Conference on Radar Meteorology*, 2001, pp. 104–106.

[56] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanisms," *Internet RFC 2488*, 1999.

[57] I. F. Akyildiz and S. H. Jeong, "Satellite ATM networks: a survey," *IEEE Communications Magazine*, pp. 30–43, July 1997.

[58] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, August 2002.

Appendix A

RANGE-TIME AND SAMPLE-TIME

Figure A.1 illustrates the received voltage due to a periodic pulse train. For a single transmitted pulse, the range-time is defined as $\tau = 2r/c$, and the received voltage $V_r(t)$ at $t = \tau$ is due to back scatter from particles located in the resolution volume at range-time τ. For the periodic pulse train, the received voltage at the same range-time (τ) is given as $V_r(t = \tau)$, $V_r(t = \tau + T)$,....., $V_r(t = \tau + mT)$ which form a sequence of temporal samples from the identical resolution volume. For a given range-time (τ), the voltage samples $V_r(t = \tau + mT) = V_r(\tau, t_s = mTs)$ can be interpreted as regularly spaced samples along the continuous sample-time axis (t_s) [12].

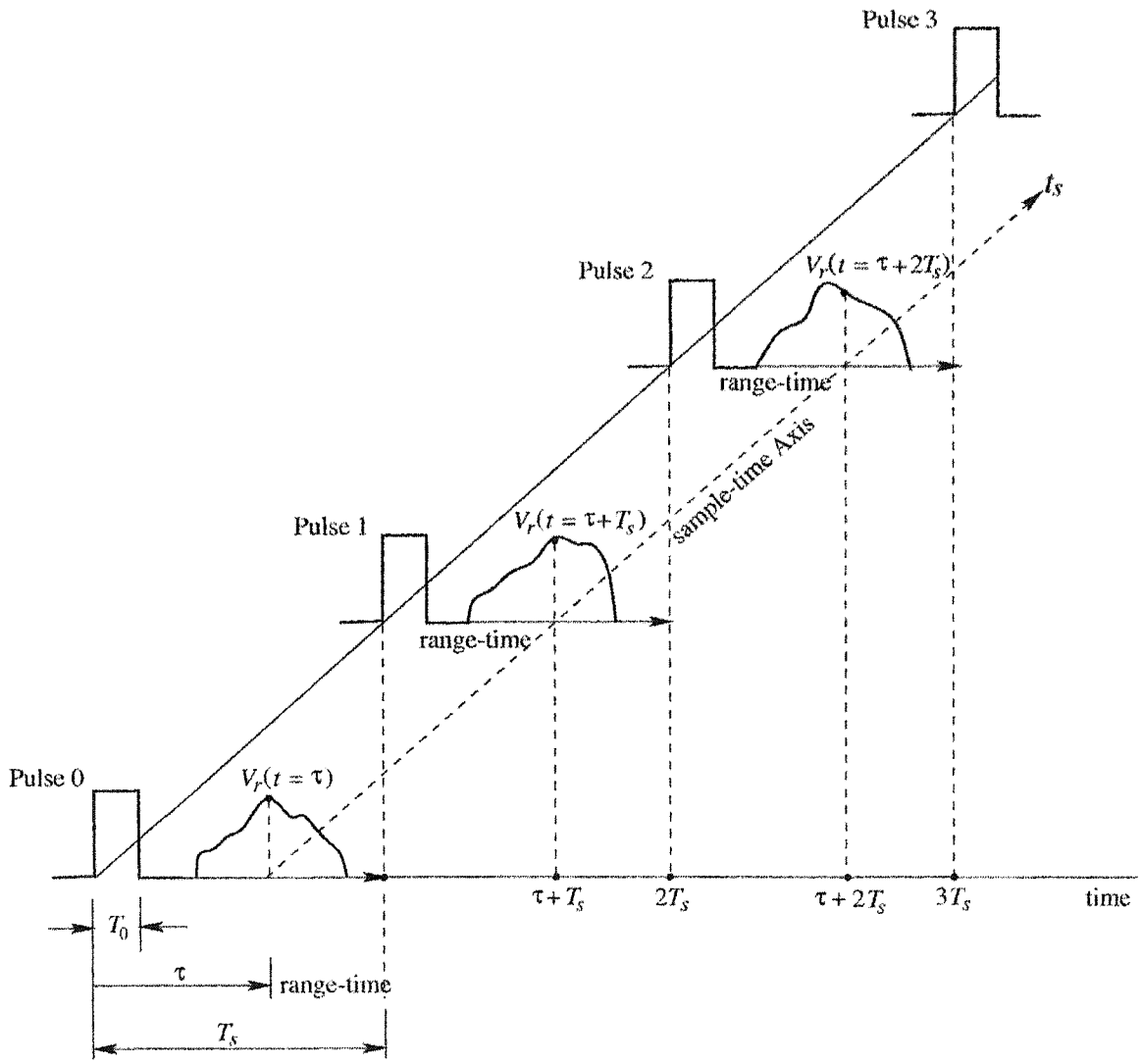
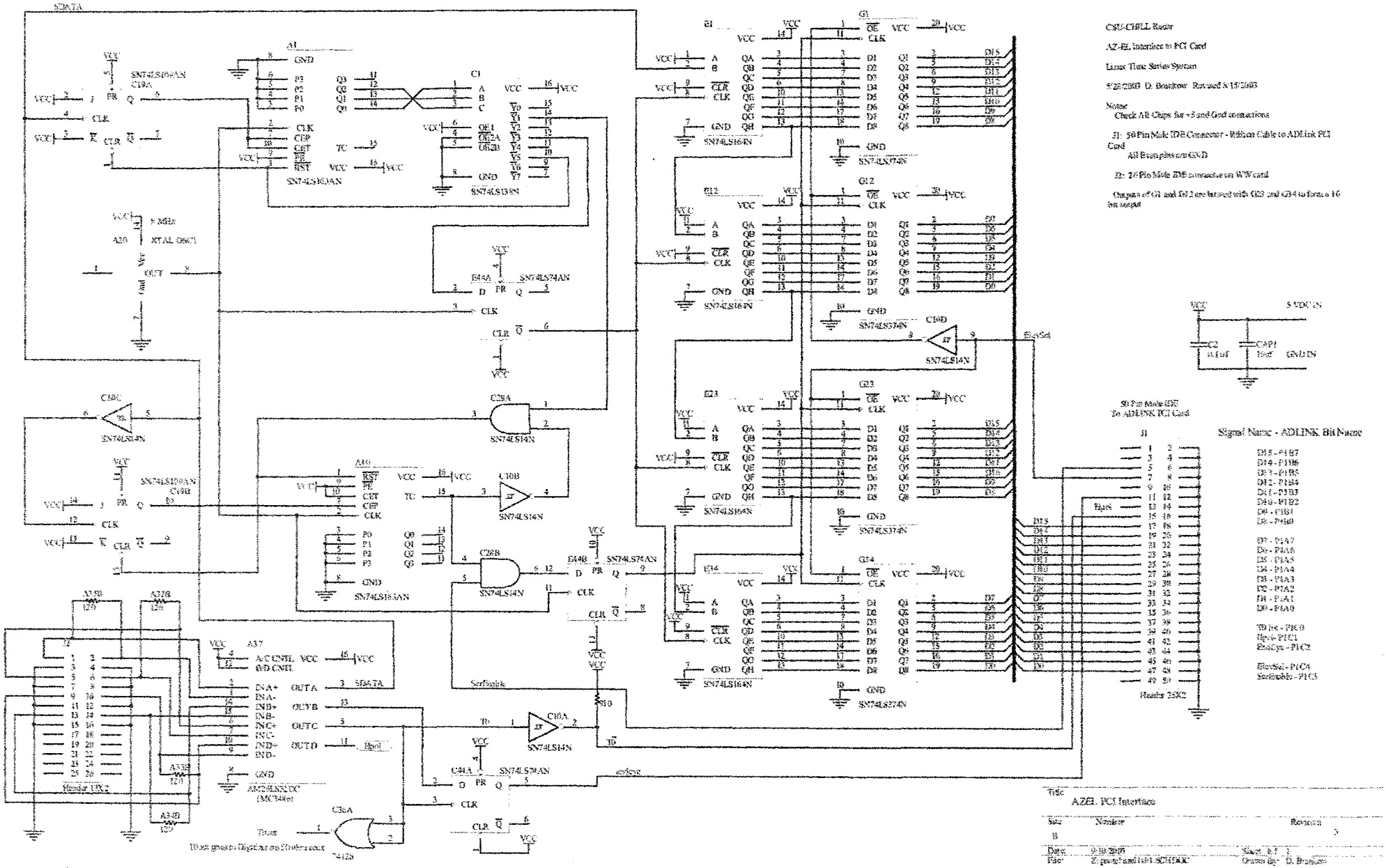


Figure A.1: Illustration showing the (continuous) range-time axis (τ) and the (discrete) sample-time axis (t_s). The pulse repetition time (PRT) is T_s.

Appendix B

AZIMUTH/ELEVATION ANGLE INTERFACE CIRCUIT



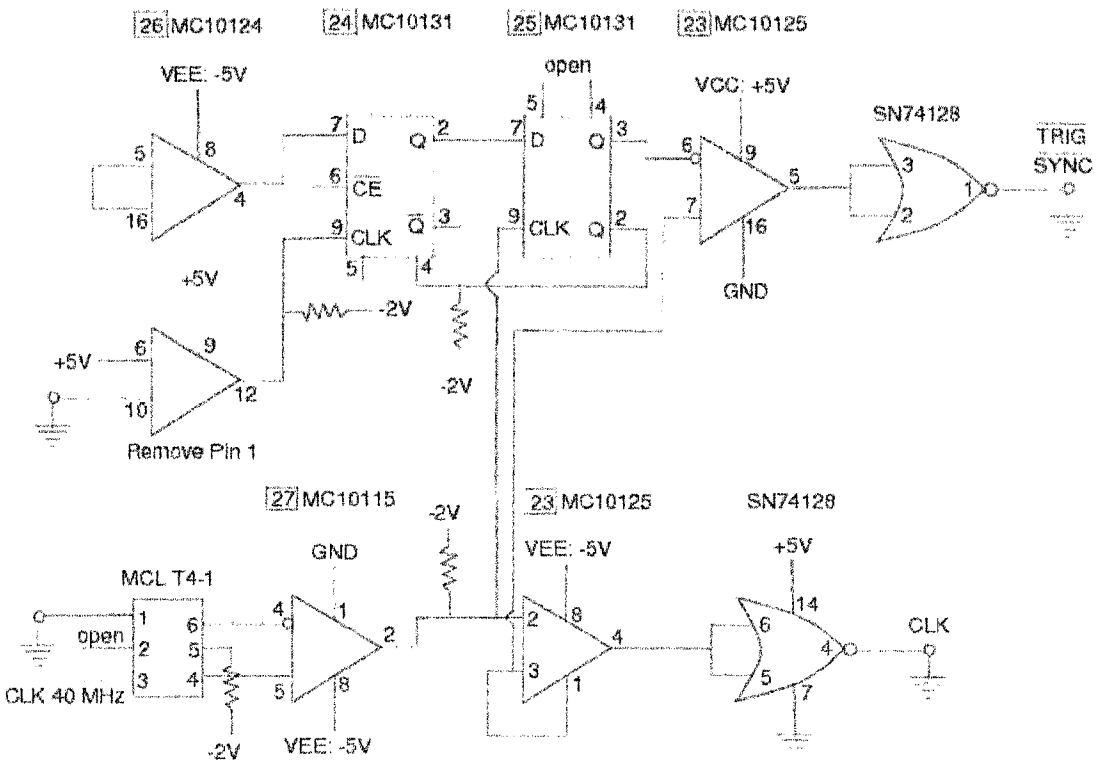


Figure B.1: (a) Circuit diagram for the conversion from the serial data type of Azimuth and elevation agles to the parallel type of data. (b) Circuit diagram for generating the external sync signal.

Appendix C

ALGORITHMS FOR ESTIMATING RADAR SIGNAL PARAMETERS

C.1 Alternating mode

Let the total number of received signal samples be $2N$ with N interlaced samples at v or h polarization states. Let $V_{vv}[2n-1]$, $V_{hv}[2n-1]$, $V_{vh}[2n]$, $V_{hh}[2n]$, $(n = 1, \ldots N)$ be the samples of the received signal as shown in Fig. C.1.

The power estimators are written as,

$$\widehat{P}_{co}^{v} = \frac{1}{N}\sum_{n=1}^{N}|V_{vv}[2n-1]|^2 \tag{C.1a}$$

$$\widehat{P}_{cxv} = \frac{1}{N}\sum_{n=1}^{N}|V_{hv}[2n-1]|^2 \tag{C.1b}$$

$$\widehat{P}_{co}^{h} = \frac{1}{N}\sum_{n=1}^{N}|V_{hh}[2n]|^2 \tag{C.1c}$$

$$\widehat{P}_{cxh} = \frac{1}{N}\sum_{n=1}^{N}|V_{vh}[2n]|^2 \tag{C.1d}$$

Two correlation estimators $\widehat{R}_{hh,vv}[1]$ and $\widehat{R}_{vv,hh}[1]$ can be written as,

$$\widehat{R}_{hh,vv}[1] = \frac{1}{N}\sum_{n=1}^{N} V_{hh}[2n](V_{vv}[2n-1])^* \tag{C.2a}$$

$$\widehat{R}_{vv,hh}[1] = \frac{1}{N-1}\sum_{n=1}^{N-1} V_{vv}[2n+1](V_{hh}[2n])^* \tag{C.2b}$$

The corresponding correlation coefficient estimates are written as,

$$\hat{\rho}_{hh,vv}[1] = \frac{\widehat{R}_{hh,vv}[1]}{[\widehat{P}_{co}^{v}\widehat{P}_{co}^{h}]^{1/2}} \tag{C.3a}$$

$$\hat{\rho}_{vv,hh}[1] = \frac{\widehat{R}_{vv,hh}[1]}{[\widehat{P}_{co}^{h}\widehat{P}_{co}^{v}]^{1/2}} \tag{C.3b}$$

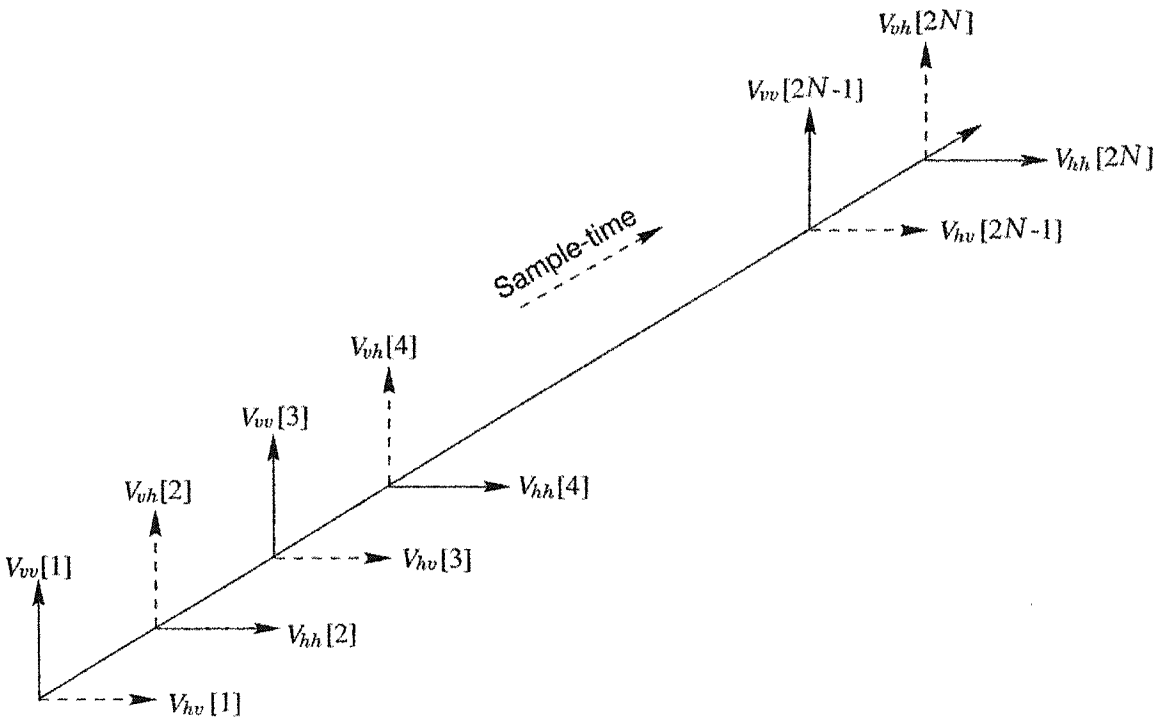


Figure C.1: Schematic diagram showing the copolar and cross-polar received signals for the alternating polarization mode with the first transmit pulse at vertical polarization.

The phase of $\widehat{R}_{hh,vv}[1]$ and $\widehat{R}_{vv,hh}[1]$ can be written as,

$$\Psi_1 = arg\{\widehat{R}_{hh,vv}[1]\} \tag{C.4a}$$

$$\Psi_2 = arg\{\widehat{R}_{vv,hh}[1]\} \tag{C.4b}$$

To calculate the velocity and the differential phase, the Ψ_1 and Ψ_2 should be transformed with an arbitrary ϕ_{rot} to allow optimal setting of the radar system phase difference (phidp) between the vertical and horizontal paths. The system phase difference comes from the differences in the transmitting Klystrons and the wave guide difference between the vertical and horizontal channels. The ϕ_{rot} is set so that the phidp starts around 80 degrees. This maximizes the amount of phase shift which can occur before phidp aliasing occurs.

$$\Psi_1' = arg\{\widehat{R}_{hh,vv}[1] \cdot (\cos(\phi_{rot}) + j\sin(\phi_{rot}))\} \tag{C.5a}$$

$$\Psi_2' = arg\{\widehat{R}_{vv,hh}[1] \cdot (\cos(\phi_{rot}) - j\sin(\phi_{rot}))\} \tag{C.5b}$$

The estimates of $\hat{\rho}[2]$ can be derived from $V_{vv}[n]$ and $V_{hh}[n]$ as follows,

$$\hat{\rho}[2] = \frac{\left[\frac{1}{N-1}\sum_{n=1}^{N-1} V_{vv}[2n+1](V_{vv}[2n-1])^*\right]}{\widehat{P}_{co}^{v}} \tag{C.6a}$$

$$\hat{\rho}[2] = \frac{\left[\frac{1}{N-1}\sum_{n=1}^{N-1} V_{hh}[2n+2](V_{hh}[2n])^*\right]}{\widehat{P}_{co}^{h}} \tag{C.6b}$$

(1) Horizontal Reflectivity (Z_h)

The reflectivity is often defined as,

$$Z_e = \frac{P_r R^2 C}{P_t G_0{}^2 G_r} \text{ in } \frac{mm^6}{m^3} \tag{C.7}$$

where P_r and P_t are the received power and the transmitted power in mW, respectively. R is the range in km, G_o is the antenna gain, G_r is the receiver gain. Here, C is defined as,

$$C = \frac{\lambda^2 2^9 \ln(2) 10^{24}}{\pi^3 |K_w|^2 h \Theta \Phi} \tag{C.8}$$

and for the CSU-CHILL radar, $10\log_{10}(C) = 287.69$. Therefore,

$$\begin{aligned} \texttt{dBZ} &= 10\log_{10}(Z_e) && \text{(C.9a)} \\ &= 287.69 - 10\log_{10}(G_o) - 10\log_{10}(G_r) - 10\log_{10}(P_t) && \text{(C.9b)} \\ &\quad +10\log_{10}(\widehat{P}^h_{co}) + 20\log_{10}(R) - 40 \\ &= 247.69 - 10\log_{10}(G_o) - 10\log_{10}(G_r) - 10\log_{10}(P_t) && \text{(C.9c)} \\ &\quad +10\log_{10}(\widehat{P}^h_{co}) + 20\log_{10}(R) \end{aligned}$$

Here, forty is subtracted because the Lassen system adopted the range as $\frac{R}{100}$ Km instead of Km. Currently, $10\log_{10}(G_o)$ and $10\log_{10}(G_r)$ are set to be 42.2 dB and ≈ 130.5 dB, respectively.

(2) $\widehat{Z}_{dr}$

In the beginning, the signal power of the horizontal and vertical receivers are calculated.

$$P_{hns} = \widehat{P}^h_{co} - P_{nsh} \tag{C.10a}$$

$$P_{vns} = \widehat{P}^v_{co} - P_{nsv} \tag{C.10b}$$

where $\widehat{P}^h_{co}$ and $\widehat{P}^v_{co}$ are the horizonatl and vertical copolar received power, respectively. P_{nsh} and P_{nsv} are the horizontal and vertical noise power, respectively. If both $\frac{P_{hns}}{P_{nsh}}$ and $\frac{P_{vns}}{P_{nsv}}$ are greater than 1.1, then the $\widehat{Z}_{dr}$ is defined as,

$$\widehat{Z}_{dr} = 10\log_{10}\left\{\frac{P_{hns}}{P_{vns}}\right\} + \texttt{zdr\_cal} \tag{C.11}$$

Otherwise $\widehat{Z}_{dr} = 0$. Here, `zdr_cal` (dB) is defined as,

$$\texttt{zdr\_cal} = \texttt{zdr offset} + (\texttt{V-transmit power} - \texttt{H-transmit power}) \tag{C.12}$$

(3) Linear Depolarization Ratio with horizontal transmission (LDR_{vh})

$$LDR_{vh} = 10\log_{10}\frac{\widehat{P}_{cxh}}{\widehat{P}^{h}_{co}} \tag{C.13}$$

(4) Linear Depolarization Ratio with vertical transmission (LDR_{hv})

$$LDR_{hv} = 10\log_{10}\frac{\widehat{P}_{cxv}}{\widehat{P}^{v}_{co}} \tag{C.14}$$

(5) Velocity (v)

$$\widehat{v} = -\frac{\lambda}{4\pi T_s}\frac{1}{2}\{\Psi'_2 + \Psi'_1\} \tag{C.15}$$

(6) Differential Phase ($\widehat{\Psi}_{dp}$)

$$\widehat{\Psi}_{dp} = \frac{1}{2}(\Psi'_2 - \Psi'_1) \tag{C.16}$$

(7) $|\hat{\rho}_{hv}|$

$$|\hat{\rho}_{hv}| = |\hat{\rho}_{hh,vv}[0]| \tag{C.17a}$$

$$= \frac{|\widehat{\rho}_{hh,vv}[1]|}{|\widehat{\rho}[2]|^{0.25}} \tag{C.17b}$$

(8) Normalized magnitude of R1 (sqi)

$$sqi = \frac{\left|\frac{R_{hh,vv}[1]+R_{vv,hh}[1]}{2}\right|}{\sqrt{\widehat{P}^{h}_{co}\widehat{P}^{v}_{co}}} \tag{C.18}$$

C.2 Hybrid mode

According to the Fig. C.2, the power estimators in the v- and h- channels can be written as,

$$\widehat{P}_v = \frac{1}{N}\sum_{n=1}^{N}|V_{vv}[n]|^2 \tag{C.19a}$$

$$\widehat{P}_h = \frac{1}{N}\sum_{n=1}^{N}|V_{hh}[n]|^2 \tag{C.19b}$$

The crosscorrelation estimators between V_h and V_v with lag 0 is,

$$\widehat{R}_{v,h}[0] = \frac{1}{N}\sum_{n=1}^{N}V_{vv}[n]V^*_{hh}[n] \tag{C.20b}$$

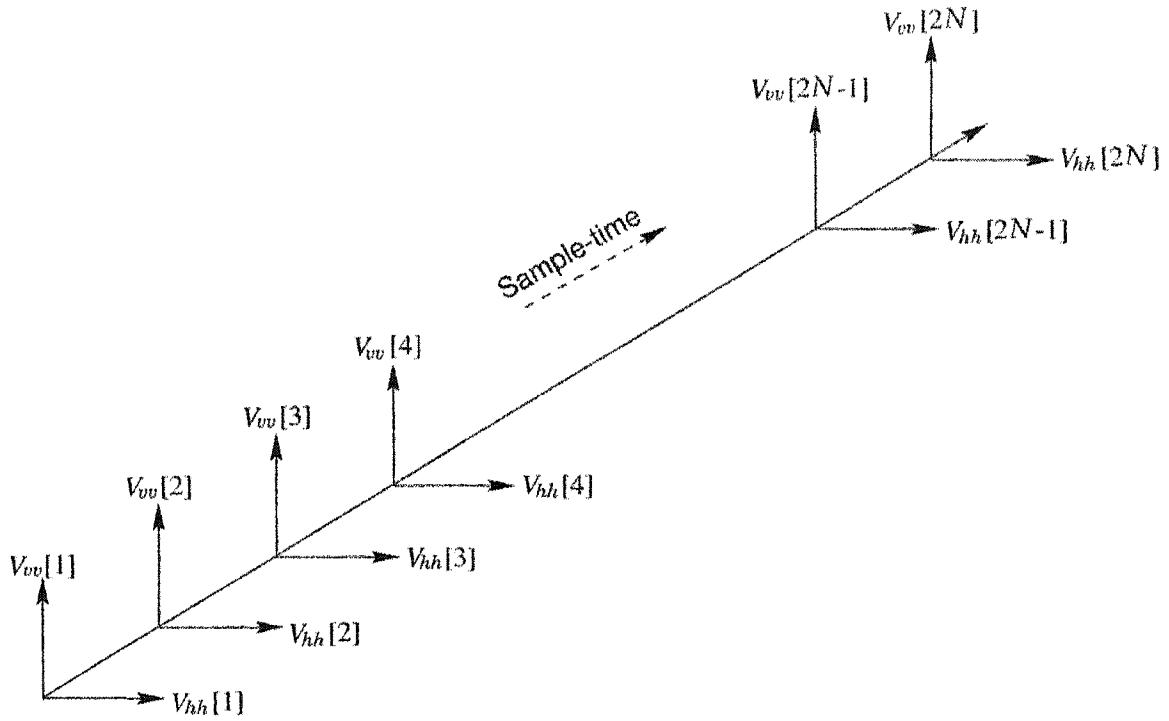


Figure C.2: Schematic diagram showing the horizontal and vertical components of the received signals in the hybrid mode.

The autocorrelation estimators with lag 1 and lag 2 can be written as,

$$\widehat{R}[1] = \frac{1}{N-1}\sum_{n=1}^{N-1} V_{hh}[n+1]V_{hh}^{*}[n] \tag{C.21a}$$

$$\widehat{R}[2] = \frac{1}{N-2}\sum_{n=1}^{N-2} V_{hh}[n+2]V_{hh}^{*}[n] \tag{C.21b}$$

(1) Horizontal Reflectivity (Z_h)

The horizontal reflectivity is defined as the same with the alternating mode.

(2) $\widehat{Z}_{dr}$

The differential reflectivity is defined as the same with the alternating mode.

(3) Velocity (v)

$$\widehat{v} = -\frac{\lambda}{4\pi T_s} arg[\widehat{R}[1]] \tag{C.22}$$

(4) Differential Phase (Ψ_{dp})

$$\widehat{\Psi}_{dp} = arg[\widehat{R}_{v,h}[0] \cdot (\cos(\phi_{rot}) + j\sin(\phi_{rot}))] \tag{C.23}$$

(5) $|\hat{\rho}_{vh}|$

$$|\widehat{\rho}_{vh}| = \frac{|\widehat{R}_{v,h}[0]|}{(\widehat{P}_h \widehat{P}_v)^{1/2}} \tag{C.24}$$

(6) Normalized spectral width (w)

There are two equations to calculate the spectral width ($\hat{\sigma}_v$).

$$\hat{\sigma}_v = \frac{\lambda}{2\pi T_s \sqrt{2}} \left[\ln \left| \frac{R(0)}{R(1)} \right| \right]^{1/2} \tag{C.25}$$

$$\hat{\sigma}_v = \frac{\lambda}{2\pi T_s \sqrt{6}} \left[\ln \left| \frac{R(1)}{R(2)} \right| \right]^{1/2} \tag{C.26}$$

Normalized spectral width (w) is written as,

$$w = \frac{\hat{\sigma}_v}{(\frac{\lambda}{2T_s})} \tag{C.27a}$$

$$= \frac{1}{\pi\sqrt{2}} \left[\ln \left| \frac{R(0)}{R(1)} \right| \right]^{1/2} \tag{C.27b}$$

(7) Normalized magnitude of R1 (sqi)

$$sqi = \frac{|\widehat{R}[1]|}{\widehat{P}_h} \tag{C.28}$$

Appendix D

SCALING FOR DISPLAYING CHILL FORMAT DATA WITH PPI PROGRAM

It is required to scale the calculated radar signal data moment into integers which ranges from 0 to 255. This scaled data are retransmitted with the multi-casting protocol to the end users. The offset and scaling values are different for each parameter and they depend on the way the ppi has been programmed to display the parameters. The offset and scaling for the different values, which can be derived from ppi_ctl.c, are specified as follows. Here, calculated values stand for the values which come out of the calibration.

- Scaled Z

  $Scaled\_Z = (Calculated\_Z + 32) \times 2$

- Scaled Zdr

  $Scaled\_Zdr = (Calculated\_Zdr + 3.01) \times (64/3.01)$

- Scaled LDR (available for LDRh and LDRv)

  $Scaled\_ldr = (Calculsted\_ldr + 48.165) \times (255/48.165)$

- Scaled Ψ_{dp}

  $Scaled\_Phi = Calculated\_Phi * (127/180) + 128$

- Scaled Rho

  $Scaled\_rho = (Calculated\_rho - 0.3) \times (200/0.7)$

- Scaled V

  $Scaled\_V = Calculated\_V \times (127/(0.5 \times Nyquist\_interval)) + 128$

- Scaled W

  $Scaled\_W = (Calculated\_W \times (255) \times (WIDTH\_VALUE) \times 10)/Nyquist\_interval$

- Scaled SQI

  $Scaled\_sqi = Calculated\_sqi \times 255$