ICICS-PCM 2003
15-18 December 2003
Singapore

802.11a MAC Layer: Firmware/Hardware Co-Design

J. H. Yeong X. M. Rao M. R. Shajan Q. Wang C. Y. Lin, Jason X. H. Qu

Institute for Infocomm Research
20, Science Park Road, #02-34/37
TeleTech Park, Singapore Science Park II
Singapore 117674
(65) 6872 9030
{yeongjh, raoxm, mrshajan, wangqi, lincy, quxh}@i2r.a-star.edu.sg

Abstract

Embedded wireless communication devices nowadays are getting more and more sophisticated and running at higher speeds than ever before. In order to achieve the flexibility of design and shorten the time-to-market, a firmware/hardware co-design approach is being adopted more often. The IEEE 802.11a MAC layer design conducted at the Institute for Infocomm Research uses this concept to realize the full potential of the protocol and at the same time provide the flexibility for future upgrade. This paper explains in detail the process of firmware/hardware co-design and the rationale behind our approach.

Abbreviations

3G	3rd Generation
ACK	Acknowledgement
AMBA	Advanced Microprocessor Bus Architecture
AP	802.11a WLAN Access Point
API	Application Programming Interface
ARM	Advanced RISC Machine
ASIC	Application-Specific Integrated Circuit
BSS	Infrastructure Basic Service Set
CM	Core Module
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	DCF Inter Frame Space
DPRAM	Dual Port RAM
FPAG	Field Programmable Array Gate
FPGA	Field Programmable Gate Array
GHz	Giga Hertz
IBSS	Independent Basic Service Set
IEC	International Engineering Consortium
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
LINUX	Open Source Operating System
LLC	Logical Link Layer
MAC	Medium Access Control
Mbps	Mega Bits Per Second
MIB	Management Information Base
MMS	Multimedia Messaging Service
MODEM	Modulator and Demodulator
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnect
PC	Personal Computer
PCF	Point Coordination Function
PHY	Physical Layer
PS-Poll	Power Save Poll
QoS	Quality of Service
RF	Radio Frequency
RISC	Reduced Instruction-Set Computing
RTS	Request To Send
SDL	Specification and Description Language
SIFS	Short Inter Frame Space
SoC	System on Chip
STA	802.11a WLAN Station
TBTT	Target Beacon Transmission Time
TIM	Traffic Indication Map
TSF	Timer Synchronization Factor
VHDL	VHSIC Hardware Description Language
WLAN	Wireless LAN

1. Introduction

The IEEE 802.11b wireless LAN, capable of providing data rates up to 11Mbps, has recently received increasing public attention. Many well-known universities have launched the wireless campus projects where 802.11b hot spots are setup in the campus. Telecom Services Providers have joined with business houses such as StarBucks café to provide wireless LAN services at their retail premises. Referring to the recent IEC research report entitled "Seamless Mobility: The Marriage of 3G and Wi-Fi", it

0-7803-8185-8/03/$17.00 © 2003 IEEE

Apple EX1041
Apple Inc. v. Secure Comm. Techs., LLC
IPR2026-00447

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.

seems that wireless LAN will become more powerful when used in conjunction with 3G networks.

Multimedia applications such as MMS provide more interesting and interactive data services to subscribers. This results in increased customer demand for multimedia data service than for text based data service. This strong demand in multimedia data services call for higher speed mobile communications networks and devices to provide seamless and uninterrupted data flows.

The IEEE 802.11a WLAN standard is designed to achieve a maximum data rate of 54Mbps. With this higher data rate, IEEE 802.11a will surpass 802.11b in delivering multimedia content. Also, 802.11a provides twelve independent channels in the 5GHz band unlike three in the 2.4GHz ISM band for 802.11b. Thus 802.11a wireless LAN is identified by the Institute as the potential wireless technology that will gain increasing popularity in wireless communications and hence considered a focus area for research and development.

In this paper, the experience of designing the IEEE 802.11a wireless LAN Media Access Control protocol is discussed in details. In section 2, the overall operation of IEEE 802.11a WLAN will be discussed. Section 3 layouts the behavioral components of the system. The hardware architecture selection is discussed in section 4. The job partitioning process is discussed in section 5. In section 6, the conclusion about the firmware/hardware co-design approach will be given.

2. Operation of 802.11a WLAN

The logical architecture of 802.11a defines the network operation at Layers 1 and 2 of the OSI network model. Layer 2 consists of the LLC and MAC while Layer 1 consists of the PHY. 802.11a uses CSMA/CA MAC and 5GHz OFDM PHY. The 802.11a WLAN is designed to support eight different transmission rates, namely 54 Mbps, 48Mbps, 36Mbps, 24Mbps, 18Mbps, 12Mbps, 9Mbps and 6Mbps. The highest bandwidth of 54 Mbps that 802.11a provides is more than sufficient to carry multimedia data.

The 802.11a specification defines two topologies: IBSS network (also known as ad-hoc network) and BSS network. In BSS network, there are two medium access schemes, namely DCF and PCF, whereas in IBSS network, only DCF is supported. DCF is contention-based protocol, whereas PCF is contention-free protocol.

This discussion focuses on the MAC layer implementation of DCF in BSS network. The BSS network consists of an AP and multiple STAs. The STAs can only communicate to one another or to another PC in the wired network through the AP.

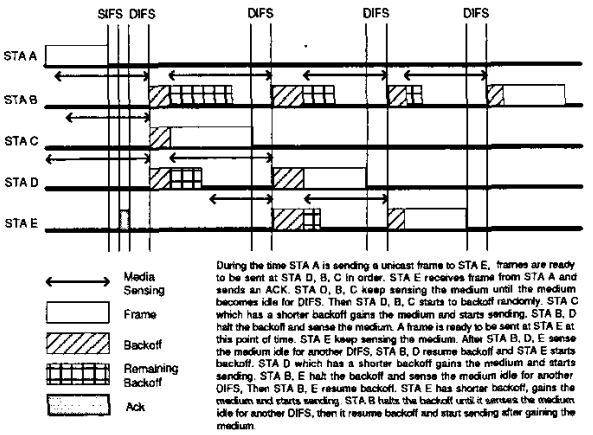

Figure 1 DCF Operation

DCF is the mandatory access protocol in 802.11a to provide contention based frame transfer. Each STA has a counter called NAV that stores the duration information of the frame that is in transmission. The STA must wait until NAV becomes zero before the next transmission attempt.

After the NAV becomes zero, a STA must detect the medium condition through PHY coordination before it can start any transmission. If the medium is idle, the STA must continue to sense the medium for a minimum period of DIFS (34μs in 802.11a). Only if the channel remains idle during DIFS, the STA is allowed to initiate a transmission. If the medium becomes busy during DIFS, the MAC implements a random backoff algorithm before the next transmission attempt to minimize the chances of collision. STA with the lowest backoff time will be given higher priority to access the medium.

The random backoff time is in the units of a slot time (9μs in 802.11a) and is determined from a uniform distribution over the interval [0, CW]. The CW is an integer within the range of values of the MIB attributes aCWmin (15 in 802.11a) and aCWmax (1023 in 802.11a). The aCWmin attribute is initially set to 15 slots and increases exponentially to a maximum of 255 slots if STA goes through successive retransmissions. This mechanism introduces more delay to backoff providing substantial transmission spacing between STAs that are attempting to transmit. Thus different CW minimizes further collisions for highly utilized network and maximizes throughput for low utilized network.

For each successful reception of a frame, the receiving STA immediately acknowledges reception by sending an acknowledgement frame (ACK) in SIFS (16μs in 802.11a). For every ACK received, the transmitting STA will reset its aCWmin to the initial value of 15.

The AP will send out Beacon frame at a constant beacon interval namely TBTT. The information in Beacon frame is used to synchronize the TSF counter in AP and STAs. The

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.

TSF counter is used for most of the MAC activities such as backoff and the calculation of NAV.

In the transmission of MAC packets, there are additional mechanisms that can be adopted. RTS/CTS mechanism is used to avoid hidden node problem. Fragmentation of MPDU can be adopted to divide a long MPDU into shorter size MPDU. This mechanism increases the probability of sending MAC packets successfully in noisy environment. Retransmission mechanism is used to reduce unicast MAC packets losses in the air.

On top of the underlying medium access protocol mentioned above, the 802.11a MAC is responsible for the MAC layer management functions. Before each STA is able to transmit data in a BSS network, it must first detect the existence of the BSS network by sensing Beacon frame. Then it must join the network and start the authentication process with the AP. After authentication is done successfully, the STA will initiate the association sequence with the AP. The AP can be configured to support either Shared Key authentication or Open System authentication. A filter list can be configured at the AP also to determine which stations are allowed to associate with the AP.

3. Behavioral Components

In this project, as the underlying medium access mechanism is the same regardless of AP or STA, we have adopted the approach to design a common platform for both the AP and the STA. With the guidelines given in the SDL in Appendix C in the 802.11a specification, we have identified the following behavioral components of the systems. The behavioral components are identified without specifying the implementation medium.

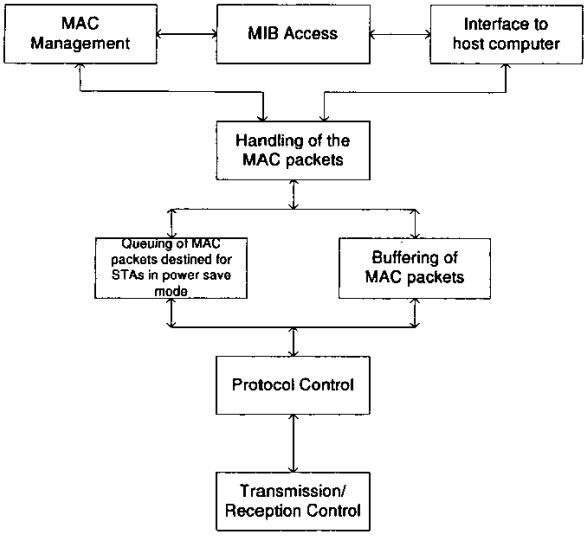

Figure 2 Behavioral Components Of 802.11a MAC Layer

Interface to the host computer. This interface links the embedded system and the host in order to transfer data packets between them and to perform wireless extension commands such as setting or getting the MIB attributes' values.

Handling of the MAC packets. For the outgoing stream, the data must be encapsulated with MAC header. And depending on the fragmentation threshold and security setting, fragmentation and encryption must be done accordingly. For the incoming stream, the duplicated MPDU must be filtered based on its sequence number and source MAC address. Then the MPDU must be decrypted and de-fragmented depending on the MIB attributes' settings. The MAC header must be de-capsulated before the data is handed over to the host. For incoming MPDU in AP, the AP has to decide based on the association list, whether to route the MPDU to another STA in its BSS network.

Queuing of the MAC packets destined for STAs in power save mode. In BSS network, AP always stays in active mode but STAs may be configured to work under power-save mode. STAs in power-save mode will wake up at a predefined time to receive the Beacon frame from AP. The AP keeps records of MAC packets destined to those STAs in the TIM element within Beacon frame. The Beacon frame is sent out at every TBTT. STAs, which receive the Beacon frame and sense that the AP has MAC packets destined for it, must send PS-Poll frame to the AP. AP will in return send one MAC packet (with the More-Data bit within MAC header set if there are more MAC packets destined for the same STA) for every PS-Poll frame received. The STA then continues to send PS-Poll frames until all buffered MAC packets at AP are drained.

Buffering of MAC packets. For outgoing stream, the speed at which each MAC packet can be transmitted depends on the accessibility to the medium. When there are too many STAs in a BSS network or when the medium is very noisy, MAC packets must be buffered. For incoming stream, the speed at which each MSDU can be transmitted to the host depends on how swift the host can sink data. As the WLAN is targeted for heterogeneous hosts, in order to prevent loss of MAC packets, sufficient buffering is to be provided. .

Protocol control. This component deals with some control and management frames such as RTS, CTS, ACK, Beacon and PS-Poll that are to be sent out timely or at an exact time interval. It also performs the necessary processing when such a control or management frame is received. In addition, it performs unicast MAC packets retransmission if ACK is not received after SIFS time.

Transmission/Reception control. This component provides an interface to the modem underneath for data transmission and reception. It keeps on checking the

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.

condition of the medium and manages the backoff algorithm when required.

MIB Access. Databases of system setting and system states are maintained in the embedded target. This component enables the user and the other components in the system to gain access to the databases exclusively. Specific action after some attributes are set is taken care by this component.

MAC Management. The management component provides security and privacy to the BSS network. It performs the operation flow required in order to maintain the BSS network. It monitors the power saving mode of the receiving STAs and works with the queuing component to manage MAC packets that are destined for those STAs.

4. Hardware Architecture Selection

The MAC hardware architecture selection is based mainly on the high-speed requirements of IEEE 802.11a and future ASIC upgrading. Other considerations are the ease of prototyping for modification and testing and efficient interfacing to other hardware modules.

Figure 3 shows the overall block diagram of the 802.11a WLAN prototype. The MAC sits between the NIC-Host Interface and the MODEM which controls and processes upstream and downstream payload.

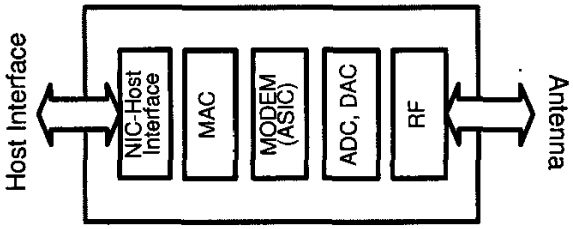

Figure 3 WLAN Prototype Hardware Blocks

A simplified MAC architecture is illustrated in **Figure 4,** which shows that all the building blocks for MAC are centered on the system backbone bus AMBA. AMBA on-chip bus is an established, open specification that serves as a framework for SoC design. It is rapidly emerging as a de facto standard for SoC design and intellectual property library development. AMBA architecture is chosen at the prototyping stage in order to ease the design, facilitate future ASIC upgrading and to enable IP reusability. Three types of ABMA devices are presented in this AMBA-based system: one Master - the ARM processor, one Decoder and three Slaves which are the MAC protocol controller, supporting peripherals and the static memory DPRAM.

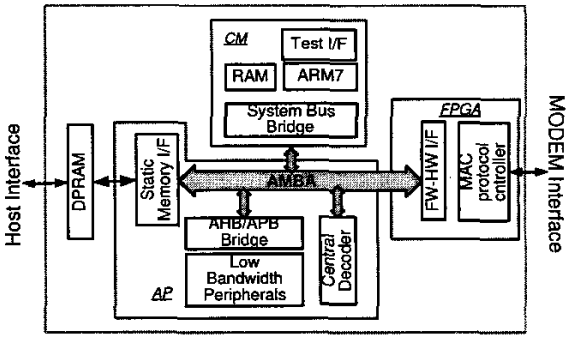

Figure 4 MAC Hardware Block

The processor is the nerve center of the MAC. The firmware is supposed to run on this processor to implement the control-intensive, less timing critical MAC functions. Through the system bus bridge, the processor presents as a Master device in the AMBA system. The local RAM memories are for instruction and data storage. The Test Interface is for firmware debugging. ARM7 is selected as the processor because of its high performance, low power consumption and low cost. It is a 32-bit RISC microprocessor optimized for cost and power sensitive embedded applications. Another important reason for choosing the ARM processor is the flexibility of design change. The processor can be easily upgraded to another higher end ARM processor cores without major changes in the design of firmware and hardware.

There is a central decoder to select one of the three slaves. The high address bus is decoded to generate the select signal for slave device and each slave may have its own local decoder to further select its local ports and memories.

The low bandwidth peripherals are the timers, serial ports, registers and interrupt controllers, etc., which support the firmware design, debugging and implementation. Through the AHB/APB bridge, they present as a single slave device on the AMBA.

The static memory interface makes the external memories accessible to the processor. The DPRAM is a dual port memory device. With one port to the host, another to the MAC processor, it provides a data and command exchanging buffer between the host and MAC.

MAC protocol controller is designed to implement the timing critical and processing-intensive MAC functions, such as encryption/decryption, CRC generation, timing synchronization, etc. It coordinates with MAC firmware through the HW-FW interface. Both the MAC protocol controller and HW-FW interface are implemented in a high speed FPGA. Xilinx XCV2000E is selected as the target device because of its high performance and high capacity. Using FPGA to implement hardware design during the prototyping stage is very flexible and convenient. The

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.

VHDL code initially targeted to FPGA can be easily ported to ASIC design.

To avoid designing everything from scratch, we have adopted two ASIC Integrator modules from ARM Limited as our basic development platform. One module is the Integrator/AP (ASIC Platform), another is the Integrator/CM (Core Module). The integrator/AP provides the AMBA backbone bus, the standard system peripherals, the central decoder and the interface to other modules. The Integrator/CM provides the ARM processor and its supporting memories. As the Integrator/AP has physical connectors for any standard AMBA devices, the third party's customized design can be easily incorporated to form a complete system. Thus, we can focus on designing the MAC protocol controller and HW-FW interface that present as a slave device in the AMBA system.

5. Job Partitioning

The goal of partitioning is to decide onto which block: firmware or FPGA, the behavioral components or parts of them should be mapped. In our approach, the criteria chosen for this decision are the speed of system response, the gate count of the FPGA, the flexibility for future upgrade and the support of different algorithms using common hardware.

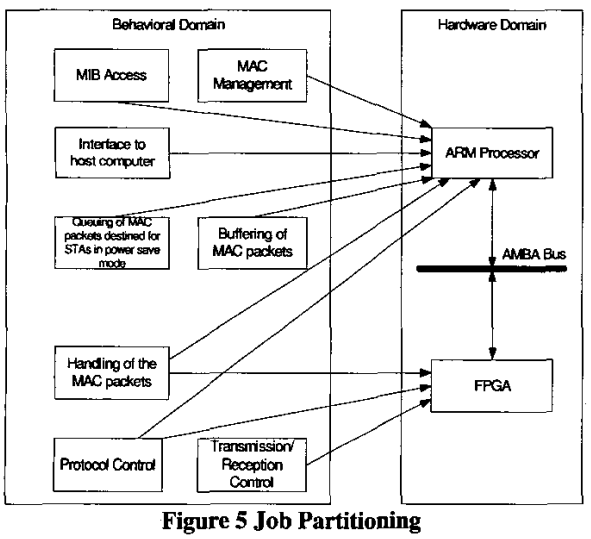

Figure 5 Job Partitioning

The interface to the host computer component is to be completely implemented in firmware in order to enjoy the flexibility of supporting different interfaces. The host interface that we are currently utilizing is PCMCIA, but with different host interface controller and re-programming of the firmware, it can be easily extended to support other interfaces such as USB or CardBus.

The encryption part of the handling of the MAC packets component is implemented in FPGA in order to offload the ARM processor that runs at a relatively low frequency of 40MHz. The remaining parts in this component are implemented in the firmware as this provides the flexibility to support different kinds of encapsulation/de-capsulation, and the flexibility for buffering the fragmented MAC packets.

As the queuing of the MAC packets destined for STAs in power-save mode is very complicated, the queuing is completely implemented in the firmware. In addition, the MAC packets in this queue must be handled together with other MAC packets that are buffered in order to provide fairness to all MAC packets destined for different STAs. Thus the buffering of MAC packets will be done in firmware too.

The protocol control component is implemented mainly in FPGA, whereas parts of it are implemented in firmware. For the RTS frame, the firmware will decide whether to use the RTS/CTS mechanism. If RTS/CTS mechanism is to be used, the firmware constructs the RTS frame and sends it together with the MAC packets. The FPGA generates and transmit CTS whenever it receives a RTS frame. Beacon frame is only sent out at AP. The firmware prepares the Beacon frame at initial power on and updates the TIM element within the Beacon frame when necessary. The Beacon is updated into the Beacon buffer in the FPGA for its transmission during TBTT. PS-Poll frame is only handled in STA. The firmware will prepare the PS-Poll frame at initial power on and update the PS-Poll frame into a PS-Poll buffer in the FPGA. The FPGA keeps track on when the STA should wake up and if it finds out from the TIM element in the Beacon frame that there are buffered frames at AP for itself. Then it sends out the PS-Poll frame accordingly. In addition, when the FPGA receives unicast MAC packet, it generates and transmit ACK frame. The CTS and ACK are generated by the FPGA as these are simple frames that need not consume lot of gates count in FPGA implementation. Whereas the firmware generates the RTS, Beacon and PS-Poll frames as the contents of these frames are dynamic and can change over time. But the sending of all these control and management frames must be done by the FPGA because the time to transmit such a reply is SIFS or at specific time interval such as TBTT. The short interval specified such as 16μs for SIFS makes it impossible to prepare and send these frames by the firmware. The same consideration applied to the retransmission of unicast MAC packets. FPGA can handle the retransmission faster than firmware once it does not receive ACK after SIFS time.

The transmission/reception component is implemented entirely in FPGA. As the continuous checking of the medium as well as the coordination between the backoff mechanism and the transmission mechanism are all time-critical, handling of this component in FPGA ensures the timeliness and better utilization of the medium.

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.

The MIB access component is implemented entirely in the firmware. Common APIs are provided in the firmware for accessing all the different attributes in MIB. As there are more than 100 MIB attributes for 802.11a WLAN, it is natural to use a general memory space as storage for all the attributes. Implementing this component in firmware also provides the flexibility for future upgrading of the MIB attributes such as additional attributes in the 802.11i enhanced security and 802.11e QoS standards.

As the MAC is configurable to manage different topology, the MAC Management component is implemented in firmware. Firmware approach enables the switching between different management flows. Flexibility in changing the size of lists that are used in MAC management such as Authentication List and Association List is another advantage. The system administrator can easily configure the lists and provide access control more effectively with respect to the condition of the medium and the number of users that are allowed to use the wireless network.

6. Conclusion

At the time of this report writing, the RF module is not ready for testing. Hence the testing and verification of the functionalities of the MAC layers design is carried out by wired back-to-back connection through baseband MODEM. A BSS network is setup and three 802.11a MAC prototypes are used; one prototype acts as AP and the other two prototypes act as STAs. The BSS network is connected to the rest of the Intranet and Internet through a PC that is configured as a firewall.

Extensive LINUX applications such as ping, ncftp, Mozilla browser, EDNA MP3 streaming server, XMMS MP3 streaming client, APACHE WEB Servers, VNC Server and Client are used to test the functionalities of the MAC prototypes.

After thorough testing, our prototype is proven to meet the functional requirements of a BSS network specified in the IEEE 802.11a MAC Layer Specification. A demonstration setup is available at the premises of Institute for Infocomm Research.

We have presented the architecture design of an IEEE 802.11a compliant Media Access Control with emphasis on firmware/hardware co-design issues. The advantages of such an approach have been highlighted. The modularity and flexibility of our design approach makes it easily adaptable to emerging WLAN and WPAN standards. As WLAN standards evolve to higher data rate systems with more stringent requirements for security and QoS, more work needs to be done to realize an optimum architecture for the MAC. We hope this report will help researchers in identifying the areas to focus on when adopting a firmware/hardware co-design approach.

References

[1] IEEE, ANSI/IEEE Std 802.11, 1999 Edition. United States Of America: IEEE Press, 1999.

[2] Bob O'Hara and Al Petrick, The IEEE 802.11 Handbook, USA: IEEE Press, 1999.

[3] Jim Geier, Wireless LANs: Implementing High Performance IEEE 802.11 Networks, USA: Sams Publishing, 2002.

[4] ARM Limited, AMBA Specification (Rev 2.0), ARM Limited, 1999.

[5] ARM Limited, Integrator/CM7TDMI: User Guide, ARM Limited, 2001.

[6] ARM Limited, Integrator/AP ASIC Development, ARM Limited, 2001.

Authorized licensed use limited to: Southern Methodist University. Downloaded on May 12,2025 at 22:01:58 UTC from IEEE Xplore. Restrictions apply.