

Software Defined Radio Prototype (II) – Implementation and Evaluation of IEEE 802.11 Wireless LAN

Takashi Shono[†] and Munehiro Matsui

Abstract

We discuss the design of an IEEE 802.11 wireless local area network (WLAN) based on our prototype software defined radio (SDR) transceiver that supports both Personal Handy-phone System (PHS) and WLAN. The most difficult problem in implementing the WLAN in this way is meeting the short inter-frame space (SIFS) requirement of the IEEE 802.11 standard. We describe the software architecture of the prototype and explain how it can process the IEEE 802.11 WLAN protocol. The hybrid programmable architecture of our SDR is a sophisticated combination of a general-purpose microprocessor, digital signal processors (DSPs), and programmable hardware. The media access control (MAC) layer functions are executed on the central processing unit (CPU), while the physical (PHY) layer functions such as modulation/demodulation (MODEM) are processed by the DSP; higher-speed digital signal processes run on a field programmable gate array (FPGA). We also describe an experimental evaluation of the prototype for IEEE 802.11 WLAN use.

1. Introduction

As mentioned in previous papers in this issue [1], [2] and other papers [3]-[5], software defined radio (SDR), whose radio functions can be changed by replacing software instead of replacing hardware, seems to be the essential technology for the next generation of mobile communications.

More and more international researchers are tackling SDR. In particular, to prove the feasibility of SDR, several organizations have developed prototype SDR units as the first step in SDR application studies [5]. We have reported an SDR unit that supports only narrow bandwidth systems (up to a few hundred kilohertz) such as PDC (Personal Digital Cellular) and PHS (Personal Handy-phone System) [6]. We fabricated an advanced SDR transceiver that uses a newly developed flexible-rate pre-/post-processor (FR-PPP) that offers improved bandwidth (> 20 MHz), and improved flexibility, to handle wireless local area network (WLAN) systems. This prototype SDR transceiver supports both PHS [7] and IEEE 802.11 [8]

WLAN (the WLAN mode can be easily expanded to IEEE 802.11b). To confirm its feasibility, we designed an IEEE 802.11 WLAN around this SDR. This paper describes the software architecture of the prototype with its distributed and heterogeneous hybrid programmable architecture consisting of a central processing unit (CPU), digital signal processors (DSPs), and PPPs and explains how it can be used to implement the IEEE 802.11 WLAN protocol. The entire system configuration and details of the hardware architecture were described in the previous paper in this issue [2].

2. Software architecture of the prototype

In the WLAN mode, direct-sequence spread-spectrum (DSSS) is selected for the physical (PHY) layer. Multi-rate transmission is enabled by two modulation schemes: binary phase shift keying (BPSK) and quadrature PSK (QPSK). The prototype offers a distributed coordination function (DCF) that supports asynchronous frame transfer services based on carrier sense multiple access with collision avoidance (CSMA/CA). DCF is also offered by many current commercial products. The access point (AP) is a portal with a bridge function to DIX Ethernet/IEEE

[†] NTT Network Innovation Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: shono.takashi@lab.ntt.co.jp

802.3. Moreover, passive scanning with a beacon frame is used to allow a station (STA) to join the basic service set. The functions not implemented at this time include wired equivalent privacy, power management, and virtual carrier sense achieved by request to send/clear to send (RTS/CTS). Note that the inter-frame space (IFS) has a variable length, as described later.

2.1 Functional allocation of communication tasks

System performance depends to a large extent on how the critical functions such as modulation/demodulation (MODEM) processing and protocol sequence processing are assigned to the different processors. Since most DSPs have a special instruction set, parallel operation unit, and a memory bus optimized for the digital signal processing, they can achieve higher digital signal processing speeds than CPUs with the same clock speed and the same power consumption. Accordingly, the prototype has a multiprocessor

architecture like other SDRs [9]; protocol sequence processing and equipment control are done by the CPU, and digital signal processing is done in the DSPs. This approach was used in our previous SDR unit [6]. The distributed processing for PHS mode in both prototypes is shown in Fig. 1. Figure 2 shows the corresponding software configuration of the prototype in WLAN mode. We used VxWorks because the WLAN protocol has far stricter requirements in terms of interrupt response time than PHS. This real-time operating system (OS) also allowed optimization of the function allocation. Our previous SDR unit used Solaris, so it was unable to achieve adequate inter-process communication. VxWorks offers significantly shorter interrupt times, so it allows finer allocation of tasks. Future work includes optimizing the SDR design by examining other factors that influence the hardware and software configurations such as interrupt response time, context switching interval, bus bandwidth, and the control of shared memories (sem-

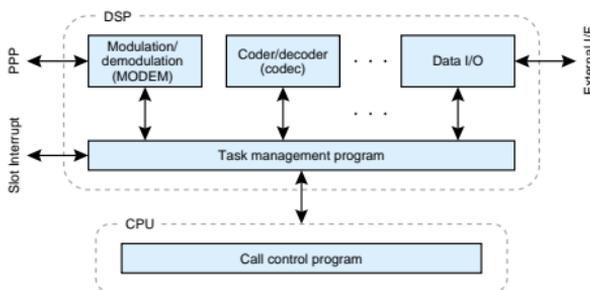


Fig. 1. Distributed processing of PHS mode.

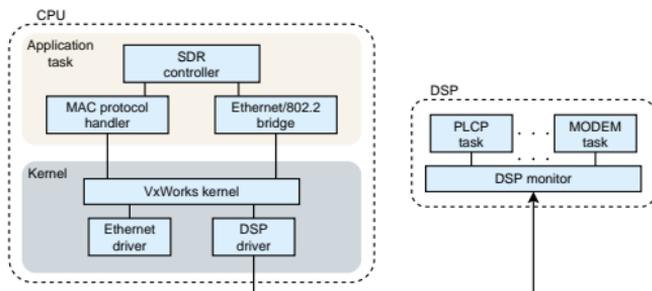


Fig. 2. Software configuration of WLAN mode.

aphore processing).

In implementing the various IEEE 802.11 functions on the prototype, we assumed that the media access control (MAC) layer functions would be executed on the CPU while the PHY layer functions such as MODEM would be processed by the DSPs. The PPP is used to perform high-speed processes that cannot be run on the CPU or DSP. For instance, the spreading process for DSSS incurs small loads, and so can be implemented on a DSP. However, detecting the correlation peak in real time at the despreading stage is beyond the ability of current DSPs, so it is done as a pre-process in the PPP.

2.2 Resolving the SIFS issue

When thinking about the PHY/MAC processes of IEEE 802.11, its short IFS (SIFS) imposes the most severe requirements in terms of response time. SIFS in the DCF mode represents the time available for completing high priority operations such as the acknowledgement (ACK) response to MAC protocol data unit (MPDU) reception and the CTS response to RTS reception. The SIFS period is defined for each PHY specification (10 μ s in DSSS), and meeting the value specified imposes very severe demands on any hybrid programmable architecture. In particular, the context switching interval of a real-time kernel is usu-

ally of the order of milliseconds and the CPU may fail to complete a task if the interval is too short.

If the functional allocation for the CPU and DSP is ideal, the delay elements of the SIFS process are as shown in Fig. 3, which gives an example of the sequence from the PHY layer convergence protocol (PLCP) protocol data unit (PPDU) reception to ACK response. SIFS is the time from the reception of the last PPDU symbol to the transmission of the first symbol of the response PPDU. The period from PPDU reception in the radio frequency (RF) part to ACK-PPDU transmission is defined as SIFS in this paper.

The delay of the processes described in the time line in Fig. 3 can be decreased to some extent by using parallel processing. The prototype supports parallel processing because it offers a high-speed interrupt handler, direct memory access, and shared memory and input/output (I/O).

To complete the testing of the prototype in the shortest period of time, we restricted ourselves to using only commercial hardware. This imposed several restrictions on system performance. The key problem was satisfying the SIFS specified in IEEE 802.11. We found that the VERSA Module European Bus (VME) bus did not offer adequate performance: 40 μ s was needed simply to transmit a 1500-byte Eth-

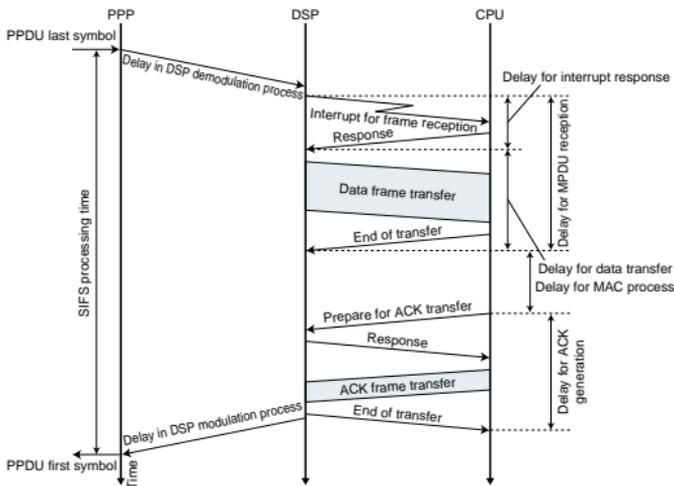


Fig. 3. Delay elements of the SIFS process.

ernet frame between the DSP and CPU. Moreover, the CPU interrupt response time was not short enough. VxWorks running on the CPU yielded an interrupt response time of 3 μ s, an appreciable fraction of the SIFS period of IEEE 802.11 (10 μ s).

Due to the above problems, ACK frames cannot be transmitted within the original SIFS period which means that frame acknowledgement fails (ACK timeout). Accordingly, to allow the performance of the prototype to be tested, we slightly modified the software to use a longer IFS. To be more precise, each IFS value was multiplied by the same integer value.

The relationship between the IFS multiplier M and each IFS value is as follows.

$$\text{SIFS}' = M \times \text{SIFS}$$

$$\text{DIFS}' = M \times \text{DIFS} = M \times (\text{SIFS} + 2 \times \text{aSlotTime})$$

$$\text{EIFS}' = \text{SIFS}' + \text{DIFS}' + (8 \times \text{ACKSize}) \\ + \text{aPreambleLength} + \text{aPLCPHeaderLength}$$

SIFS, DIFS: standard IFS value of IEEE 802.11
SIFS', DIFS', EIFS': IFS value of this prototype
aSlotTime, ACKSize, aPreambleLength, aPLCPHeaderLength:
please see IEEE 802.11 [8]

SDR with the hybrid programmable architecture can provide WLAN support by selecting a suitable value of M . Note that the slot time and ACK time-out also change as the IFS values change.

The relationship between the IFS multiplier and WLAN mode throughput is shown in Fig. 4 (the MAC frame is unicast without fragmentation). The throughput means the maximum data transmission rate of MPDU assuming that the wireless channel has no bit errors or congestion. When M is 10, throughput is about 70% of the original performance.

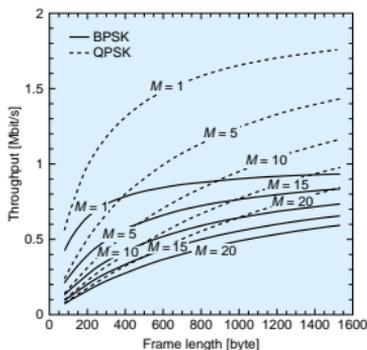


Fig. 4. Relationship between IFS multiplier and throughput for BPSK/QPSK for $M = 1, 5, 10, 15, 20$.

3. Experimental evaluation of the SDR prototype

To evaluate the WLAN performance of this prototype, we examined both the PHY and MAC layers. The PHY layer performance was described in the previous paper in this issue [2]. The MAC layer examination analyzed the throughput and DSP load.

After implementing the WLAN mode on the prototype, we found that M should be at least 10. The main barrier to supporting IEEE 802.11 on this prototype is the context switching interval of the real-time OS. The context switching interval of VxWorks can be reliably reduced to 200 μ s (10 ms is the default value), which supports the IFS-extended version of the IEEE 802.11 standard. We also found that VxWorks was sensitive to the system clock because its internal timer is triggered from the system clock. The 5-kHz system clock on the CPU board used creates significant jitter (399 μ s). We determined that by extending SIFS to 100 μ s, the default value of this prototype, we can fully support CSMA/CA operation.

We measured throughput across a one-to-one cable connection (one AP and one STA), and across a one-to-two cable connection (one AP and two STAs).

3.1 One-to-one throughput characteristics

First of all, we examined the one-to-one throughput characteristics. The measurement environment is shown in Fig. 5(a). Packets were generated and received by connecting LAN analyzers via Ethernet to SDR sets; DIX Ethernet frames were used. At the SDR transmission set, the Ethernet frames received from the Ethernet interface (I/F) were converted into IEEE 802.11 frames without a frame check sequence (FCS), and then transmitted to the DSPs via the VME bus. The PLCP frames were first generated by adding FCS, the PLCP header, and PLCP preamble in the DSP and then writing them into the first-in first-out (FIFO) buffer in the DSP board. The FIFO output was passed to the PPP for subsequent hardware processing. The reverse process was done in the reception set, and the Ethernet frames were analyzed on the LAN analyzer.

The client PCs connected to the Ethernet I/F of the SDR sets lie in the same subnet. The STA converts the multicast frames into unicast frames and sends them to the AP for multicast release. On the downlink, packets with local (AP) MAC addresses are terminated at the AP; packets with client (STA-side) MAC addresses are sent to the air I/F. Packets with other MAC addresses are dropped at the AP. Packets

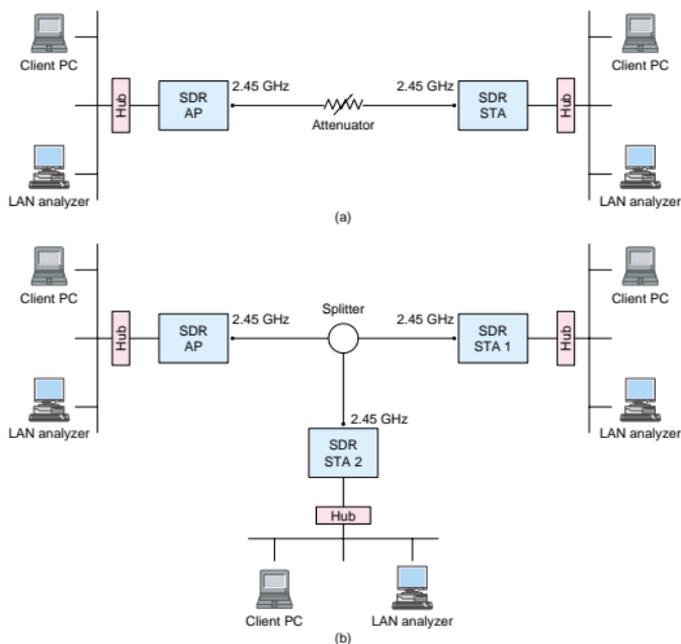


Fig. 5. System for measuring MAC layer performance. (a) one-to-one communication. (b) one-to-two communication.

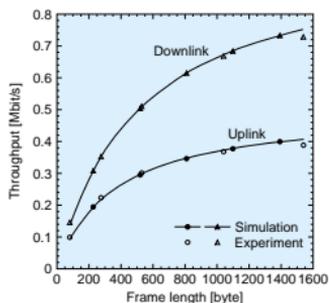
received at the STA that have local (STA) MAC addresses are terminated in the STA; packets with client (STA-side) MAC addresses are sent to the Ethernet I/F, packets with other MAC addresses are dropped at the STA. On the uplink, the STA terminates those packets that have local (STA) MAC addresses; packets with other MAC addresses are sent to the air I/F. The AP terminates those packets that have local (AP) MAC addresses; packets for other client (STA-side) MAC addresses are sent to the air I/F. Packets for other MAC address are sent to the Ethernet I/F.

Figure 6(a) shows the measured throughput when transmitting multicast frames over uplinks and downlinks. Computer simulation results are also shown for reference. The contention window (CW) size used in the computer simulation was set to 17. This value represents the average of the initial CW size because the one-to-one connection prevented frame congestion and collision. For multicast transmission over the

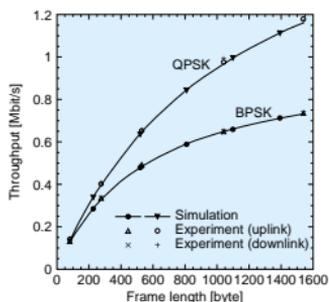
uplink, because channel contention occurs between the frame re-transmitted by the AP and the next frame of STA, the average initial CW size is about 10. Therefore, in the computer simulation of the uplink, the CW size was set to 10 assuming that frame collision did not occur. Figure 6(b) shows the measured throughput when transmitting unicast frames over uplinks and downlinks, with BPSK and QPSK. Moreover, Fig. 6(c) shows the throughput for various IFS multipliers. From Fig. 6(c) we can see that WLAN operation was achieved regardless of the IFS multiplier value. Figure 6 indicates that the SDR prototype offers throughput characteristics that closely match the results of computer simulation.

3.2 One-to-two throughput characteristics

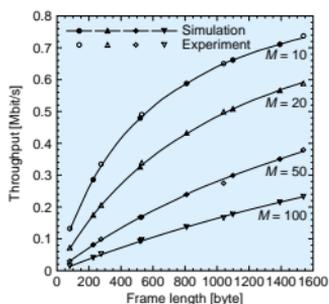
Next, we assessed the one-to-two throughput characteristics. The measurement environment is shown in Fig. 5(b). To confirm CSMA/CA operation in WLAN mode, we measured the throughput while



(a)



(b)



(c)

Fig. 6. Throughput characteristics. (a) multicast frame transfer over uplink and downlink for BPSK and $M=10$. (b) unicast frame transfer over uplink and downlink for BPSK/QPSK and $M=10$. (c) unicast frame transfer for various IFS multipliers and BPSK.

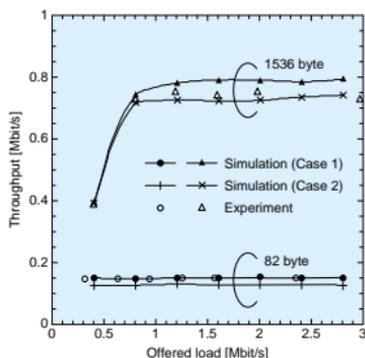


Fig. 7. Throughput characteristics vs. traffic load for unicast frames transfer with BPSK and $M=10$.

varying the offered load. Throughput was measured for two frame lengths: 82 bytes and 1536 bytes. Figure 7 shows the measured throughput across the AP with equal traffic from each STA. The horizontal axis shows the total traffic from STA 1 and STA 2, and the vertical axis shows the MAC frame throughput as received by AP. The computer simulations considered two cases. In Case 1, the local STA timer had no jitter and perfect slot synchronization was achieved between STAs. In Case 2, frame collision occurred when STA 1 and STA 2 selected adjacent slot numbers when performing backoff. Such a situation can occur in a real system because STA slots are asynchronous due to the timer jitter of the OS (see Section 2). One of the factors causing the discrepancy between the measured and Case 1 plots is this jitter. Moreover, when the traffic exceeds the maximum rate of the wireless channel, packets are dropped due to buffer overflow, which causes the throughput to saturate. Short frames (82 bytes) yield low throughput due to the large overhead in the PLCP sublayer, as can be understood from Fig. 6(b). Therefore, the throughput vs. offered traffic characteristics of this prototype agreed well with the computer simulation results, and we could confirm normal CSMA/CA operation.

3.3 DSP load characteristics

Finally, we investigated the characteristics of DSP loads, which we measured by computing the number of fetch cycles of each task. Transmission/reception processes were executed in different DSPs. The DSP

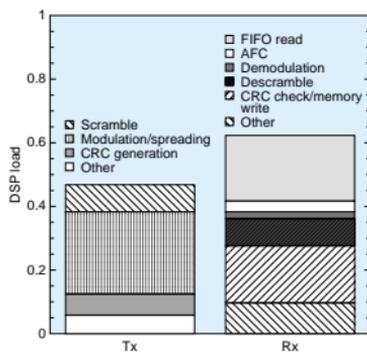


Fig. 8. DSP load characteristics.

load on the transmitting side was measured when four of the symbols constituting the MPDU underwent DQPSK modulation. The DSP load on the receiving side was measured when demodulating one symbol. Figure 8 shows the loads. Transmission loads were heavier than receiving loads because the Tx DSP could transmit MPDU while reading PLCP preamble and PLCP header, whereas the Rx DSP read data from PPP symbol by symbol. The maximum DSP load was 60%, which means that the DSPs can handle IEEE 802.11 signal processing.

4. Conclusion

We have fabricated a prototype SDR transceiver that supports both PHS and IEEE 802.11 WLAN. This paper reviewed our SDR software design methodology with distributed and heterogeneous hybrid programmable architecture that can achieve IEEE 802.11 WLAN operation. The keys to achieving this were elucidated: allocation of signal processing functions to processors and reconfigurable hardware (CPU, DSPs, and PPP), and IFS extension for full support of CSMA/CA operation, which is a fundamental MAC layer protocol of IEEE 802.11. The latter resolves the problem caused by the slow bus between the processors, interrupt response time of each processor, and context switching interval of the real-time OS. No previous report has described an SDR that supports several wireless communication systems with quite different bandwidths and communication protocols. Our experiments performed to verify the operation of WLAN mode agreed well with

theoretical results and computer simulation results.

We found that to fully support the IEEE 802.11 protocol, which has severe time-critical requirements, a dedicated programmable hardware chip is required.

We are now developing a terminal-size SDR transceiver to meet all the requirements of certain existing wireless standard specifications based on this prototype's development results.

References

- [1] K. Uehara, K. Araki, and M. Umehira, "Trends in Research and Development of Software Defined Radio," *NTT Technical Review*, Vol. 1, No. 4, pp. 10-14, 2003.
- [2] H. Shiba, Y. Shirato, H. Yoshioka, and I. Toyoda, "Software Defined Radio Prototype (I) – System Design and Performance Evaluation," *NTT Technical Review*, Vol. 1, No. 4, pp. 15-23, 2003.
- [3] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, Vol. 33, No. 5, pp. 26-38, 1995.
- [4] T. Shono, K. Uehara, and S. Kubota, "Proposal for system diversity on software defined radio," *IEICE Trans. Fund.*, Vol. E84-A, No. 9, pp. 2346-2358, 2001.
- [5] W. Tuttlebee, "Software Defined Radio – Origins, Drivers and International Perspectives," John Wiley & Sons, 2002.
- [6] Y. Suzuki, K. Uehara, M. Nakatsugawa, Y. Shirato, and S. Kubota, "Software radio base and personal station prototypes," *IEICE Trans. Commun.*, Vol. E83-B, No. 6, pp. 1261-1268, 2000.
- [7] Personal Handy-phone System, RCR STD-28, 1995.
- [8] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, ANSI/IEEE Std 802.11, 1999.
- [9] D. Efsthathiou, L. Fridman, and Z. Zvonar, "Recent developments in enabling technologies for software defined radio," *IEEE Commun. Mag.*, Vol. 37, No. 8, pp. 112-117, 1999.

Takashi Shono

NTT Network Innovation Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Keio University, Yokohama, in 1996 and 1998, respectively. In 1998, he joined the NTT Wireless Systems Laboratories, Yokosuka, Japan. Since then, he has been engaged in the research and development of software defined radio (SDR) systems, and synchronization and interference cancellation techniques for orthogonal frequency division multiplexing (OFDM) and multi-carrier code division multiple access (MC-CDMA). He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and IEEE. He received the Young Investigators Award from IEICE, and the 18th Telecom System Technology Award from the Telecommunications Advancement Foundation in 2002 and 2003, respectively.



Munehiro Matsui

NTT Network Innovation Laboratories.

He received the B.E. and M.E. degrees in information engineering from Nagoya University, Nagoya, in 1998 and 2000, respectively. In 2000, he joined NTT Network Innovation Laboratories, Yokosuka, Japan. Since then, he has been engaged in the research and development of SDR systems and OFDM systems. He is a member of the IEICE.

