Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2014, Article ID 394376, 19 pages
http://dx.doi.org/10.1155/2014/394376

Research Article

A-LNT: A Wireless Sensor Network Platform for Low-Power Real-Time Voice Communications

Yong Fu,[1] Qiang Guo,[2] and Changying Chen[3]

[1] *Shandong Province Key Laboratory of Computer Network, Shandong Computer Science Center, Jinan 250014, China*
[2] *School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan 250014, China*
[3] *Information Research Institute of Shandong Academy of Sciences, Jinan 250014, China*

Correspondence should be addressed to Yong Fu; yongfu0976@gmail.com

Received 1 November 2013; Revised 4 January 2014; Accepted 9 January 2014; Published 4 May 2014

Academic Editor: Vincenzo Eramo

Combining wireless sensor networks and voice communication for multidata hybrid wireless network suggests possible applications in numerous fields. However, voice communication and sensor data transmissions have significant differences, Meanwhile, high-speed massive real-time voice data processing poses challenges for hardware design, protocol design, and especially power management. In this paper, we present a wireless audio sensor network platform A-LNT and study and discuss key elements for systematic design and implementation: node hardware design, low-power voice codec and processing, wireless network topology, hybrid MAC protocol design based on superframe, radio channel allocation, and clock synchronization. Furthermore, we discuss energy management methods such as address filtering and efficient power management in detail. The experimental and simulation results show that A-LNT is a lightweight, low-power, low-speed, and high-performance wireless sensor network platform for multichannel real-time voice communications.

1. Introduction

WSNs (wireless sensor networks) consist of random or planned placed low-power wireless sensor nodes to monitor physical or environmental parameters [1]; these nodes are usually battery powered. WSNs have advantages of low power, low cost, self-networking, and no wiring or additional power supply needed. WSNs have been widely applied in environmental monitoring, intelligent transportation, automatic control, and so on [2]. Scientists and engineers are facing new demands and challenges with the deployments of WSN systems: precise real-time personal orientation, medical monitoring systems, access WAN and data fusion, and so forth. Voice communications have a wide range of potential applications in these WSN systems, such as staff regularly patrolling and examining, medical advice and counseling, broadcast and notification, and emergency voice communications. Although the research of WMSN (wireless multimedia sensor network) [3] has been carried out for ten years, high power and bandwidth requirements limit WMSN development [4]. In present, most WMSNs platforms are based on 802.11 platforms [5–7] and powered by high-capacity batteries or extern power supply.

In the last few years, multimedia components have become common and cheap with the rapid development of MEMS (microelectromechanical systems) and mobile Internet. Meanwhile, it becomes possible to achieve low-power and low bandwidth voice communication in WSNs with compression ratio improvement and power consumption reduction [8]. However, there are several problems that must be solved in combining voice communications with traditional WSNs: first, voice communication data and WSN data are significantly different in transmission features; audio data are real-time, which is different from ordinary WSN; voice communication needs much longer duration than sensor data transmission, meanwhile it occupoes the channel in communication. Moreover, audio sensor nodes are power-hungry and occupy massive bandwidth in data transmission. Furthermore, frequent high-speed real-time transmission of audio data poses challenges for WSN radio channel management, protocol design, hardware design, and energy management.

In this paper, we present a low-power wireless audio sensor network platform, which we called A-LNT: a lightweight low-speed and low-power wireless sensor network for voice communications, while considering WSN characteristics and hardware limitations. In Section 2, we will describe some existing WASN solutions. Then, in Section 3, we will discuss the hardware realization of A-LNT including node hardware components, power-management unit, and audio coding circuit. In Section 4, we will elaborate on the design of the MAC protocol including network topology, radio channel management, clock synchronization, and network management. In Section 5, we will describe our simulation and experiment procedures followed by reports of our results. At last, we evaluate and summarize A-LNT platform.

2. Related Works

There are a few suitable protocols and platforms for WASNs (wireless audio sensor networks) [9] at present [10–14]. Gabale et al. present a TDMA- (time division multiple access-) based MAC protocol LiT [10] and implement the MAC on an 802.15.4 platform Tmote; the evaluation of LiT shows quick flow setup, low packet delay, and essentiality for real-time applications; however, the speech coder chosen is G.723.1, and the coding bit is 6.3 kbps, but the speech codec power consumption is not considered; in 2012, a Lo^3 system based on LiT was reported [15], and it showed that such system bodes well with cost and power constraints in rural regions; the audio codec is SPEEX [16] with data rate of 5.9 Kbps, and the expected lifetime is 5 days. For most WSN applications, we need longer lifetime.

Li et al. study the audio element detection method in audio sensor network [11], and audio is treated as a special sensing data. In [12], speech codecs for high-quality voice over ZigBee applications are discussed. Zhao et al. design and implement an enhanced surveillance platform with low-power WASN; three kinds of audio sensors are discussed [13]. A voice network protocol based on session initiation protocol using TDMA/TDD MAC protocol using an IEEE802.15.4 PHY is present in [14], which is suitable for voice communications in both small- and large-scale networks. Further research of this group on full-duplex voice mixer for multiuser [17] and multiuser voice communications [18] is carried on.

Most of the above solutions are based on IEEE802.15.4/ZigBee protocol and an 8-bit RF SOC CC2430/CC2530; the ZigBee protocol is complex and huge, and the full protocol stack requires more than 100 Kbytes of flash and 5 Kbytes of ROM in CC2430/CC2530.

3. Hardware Realization

A-LNT is a WSN platform. It has the typical characteristics of WSNs: low power, self-organizing network, environmental parameter monitoring, and reliable data transmitting. Meanwhile, the platform could carry real-time voice communications without affecting sensing data transmissions. There are three types of nodes in our designed network: a central node (CNODE) for network establishing and management, sensor nodes (DNODEs) which are wireless terminals placed on target position or person for environmental monitoring and physiological parameters monitoring, and audio sensor nodes (ANODEs) which are wireless sensor terminals with audio communication functions. CNODE and DNODEs are typical nodes in WSNs, and ANODE is a new type of DNODE introduced by us for voice communications. All nodes are constructed by MCU, power management unit, RF transceiver, voltage monitoring unit, sensors, and batteries. ANODE and CNODE have additional parts for audio communications: audio processing unit, display unit, and user input unit. In order to simplify hardware design and embedded software programming, we choose the same MCU and RF transceiver for all nodes.

The MCU chosen is MSP430F2618, which is a 16-bit ultra-low-power RISC MCU; the MCLK is up to 16 MHz, and the wake-up time from low-power mode to active mode is less than 1us, which is suitable for dealing with frequent audio contents. There is an 8-channel 12-bit ADC (analog-to-digital converter) with internal reference, an internal temperature sensor, and 4 USCIs (serial communication interfaces) available in the chip, which could meet sensor interface needs for most WSNs. At present, we use the internal temperature sensor and one-channel ADC for voltage monitoring; there are 6-channel ADCs that are available for additional sensors. We chose CC2500 as the RF transceiver. It works at the ISM band of 2.4 GHz to 2.4835 GHz. The maximum wireless speed is 500 Kbps, and the current consumption is 17.0 mA at RX states, 21.1 mA@0 dBm at TX states, and 400 nA at sleep states.

DNODEs and ANODEs have different supply schemes. DNODEs are connected to batteries directly. It is an ideal way to power low-cost, low-power DNODEs as no energy loss is introduced by the power management unit. However, it is not suitable for powering ANODEs as audio codec and audio amplifier require low noise and a stable power supply. A high PSRR (power supply rejection ratio) LDO is necessary for ANODEs. Directly connecting LDO to batteries will increase current consumption and reduce available battery capacity. So a high performance step down DC-DC converter TPS62203 is added to the circuit in order to improve efficiency.

In practical application, users may want to turn off the terminal equipment when they finish their work. We use a low BISS (VCEsat) transistor PBSS5320T from NXP semiconductors and a small signal PNP transistor 9014 to realize a load switch. Although PMOS (P-Channel MosFet) transistors are popular in load switch designs, we chose a BISS transistor because it is ESD insensitive and has a constant VBE about 650 mV. So the voltage measurement circuit is easy to realize. The power control circuit workflow is as follows: when the batteries are connected to the board, the BISS transistor is off; when the tact switch $S2$ is pressed, the BISS transistor is ON, then the MCU turns on $Q3$, the board works normally, and the MCU monitors the voltage between $R4$ and $R5$. When $S2$ is pressed or the batteries voltage is lower than the threshold voltage for 30 seconds, the MCU turns off $Q2$

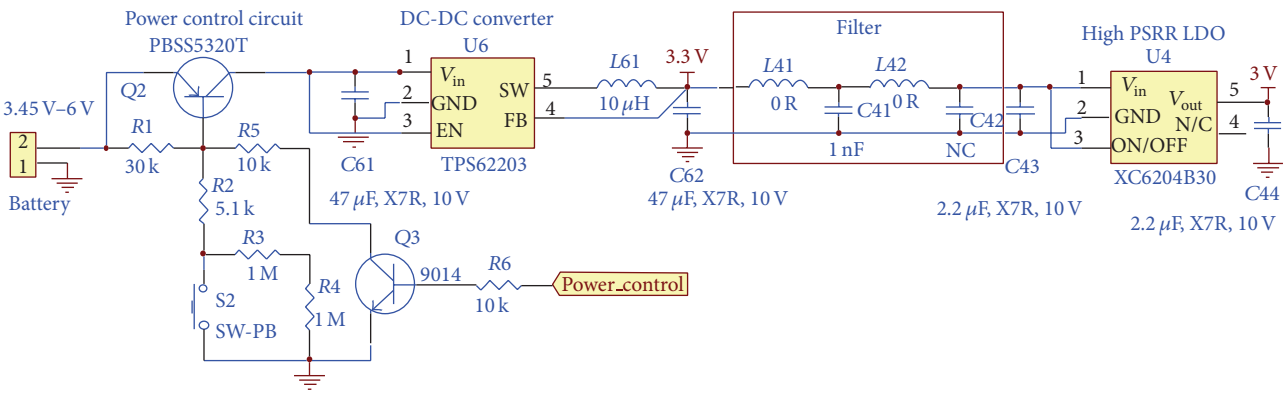

Figure 1: ANODE power management circuit.

and the board is powered down. Figure 1 shows the power management circuit schematic.

The last important part in hardware design is the audio processing unit. The audio processing unit consists of audio codec, microphone, audio amplifier, and communication interface. The audio codec algorithm should satisfy the following requirements: low power, low bit rate, and robustness for wireless communication. The CVSD (continuously variable slope delta modulation) algorithm meets all the above requirements and is an ideal solution for wireless voice communication; even the error bit ratio reaches 10%, and the MOS (mean opinion score) is greater than 3. CVSD algorithm is a simple algorithm based on PCM (pulse code modulation) algorithm, which has been widely applied in digital voice conferences and digital cordless telephones [19, 20]. The codec chip chosen is CMX649 [21], which is a low-power full-duplex codec; the typical operating current is 2.4 mA at 3.0 V, and the codec bit rate is 15.625 Kbps in the design. The codec transmits audio contents through SPI with MCU.

By now, we have finished the test-board hardware design. All test-boards in A-LNT are on a 2-layer FR-4 PCB where the board thickness is 1 mm. The DNODE test-board is a tiny board that consists of MSP430F2618, CC2500, 2 AAA batteries, and respective peripheral circuits. ANODE test-board is shown in Figure 2. The CNODE uses the same board as ANODE with different embedded software. We will discuss A-LNT MAC protocol design in the next section.

4. Protocol Design and Algorithm Realization

We divide the A-LNT protocol and software design into 3 parts: network topology, MAC protocol, and network management. Network topology is the foundation of the entire protocol design; it determines radio channel allocation and data transmission management strategies. MAC protocol is the most important part in A-LNT software design, which consists of hybrid channel access and management based on superframe, clock synchronization design, address filtering, address allocation rule, and packet priority setting. We will start from network structure design.

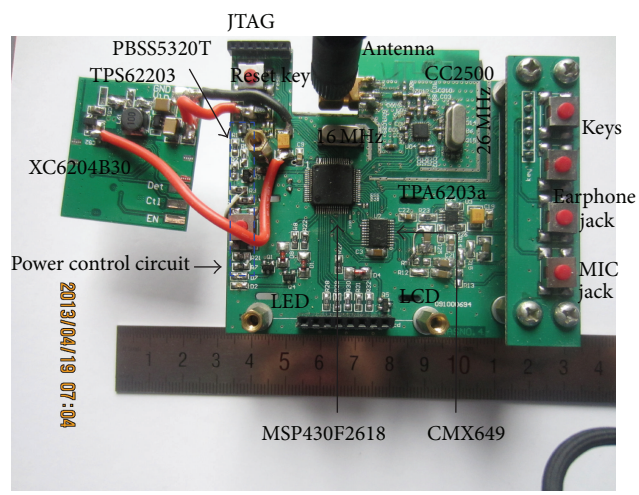

Figure 2: ANODE/CNODE test-board.

4.1. Network Structure Design. As we have mentioned above, A-LNT is based on WSN; it has typical characteristics of WSNs: low power, self-organizing network, environmental parameter monitoring, and reliable data transmitting. Meanwhile, the platform could carry out real-time voice communications without affecting sensing data transmissions. Sensing data requires reliable transmission, but latency is not critical. We ensure correct data transmission by applying the acknowledgment mechanism in WSNs. Voice communication is real-time, which needs strict clock synchronization. In addition, the real-time requirements and a large number of data transformations in voice communications determine that the network topology and protocol must be simple and efficient. Some packet loss and data error are acceptable in voice communications; acknowledgments are unnecessary and will degrade performance.

In a word, in order to meet the requirements of voice communications and sensing data transmission at the same time, the MAC protocol should be clock synchronous and ensuring two types of data noninterference. We designed the wireless network structure with considering the above factors; the network has a star topology as shown in Figure 3.

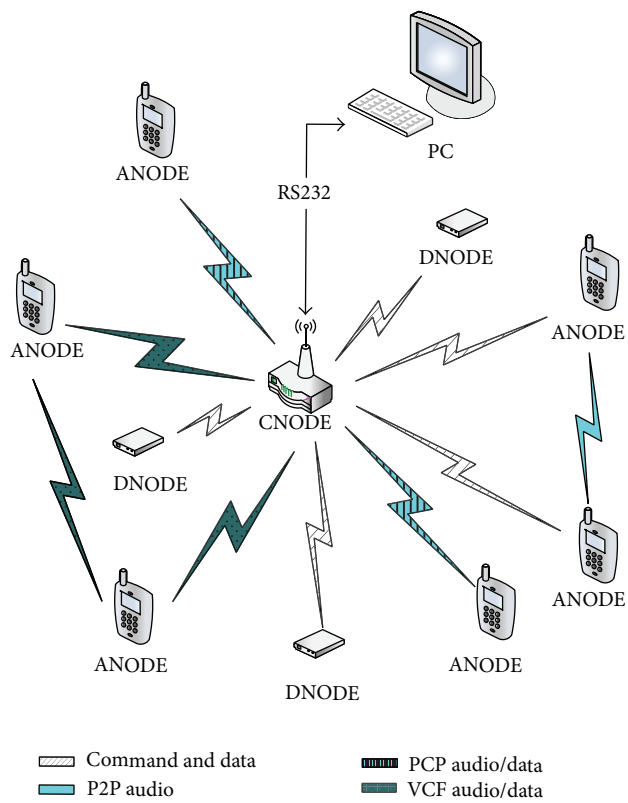

Figure 3: Network structure.

CNODE is in charge of network management, nodes management, and clock synchronizing. DNODEs measure environmental parameters and upload data to CNODE periodically. ANODEs upload sensing data in the same way of DNODEs. A-LNT supports three types of voice communications: in most conditions, voice communications between ANODEs should be peer-to-peer (P2P) in order to reduce wireless transmission pressure; if two ANODEs are too far to communicate directly, audio packets could be forwarded by CNODE (PCP); the last voice communication type is voice conference (VCF); in this mode, only one ANODE or CNODE is active while all other ANODEs are listening at one moment.

4.2. MAC Protocol Design. In order to realize the MAC protocol, we should determine clock synchronization frequency and the superframe time at first. In wireless multimedia networks, TDMA is an efficient and popular method to ensure QoS (quality of service) and maximize the use of wireless bandwidth [22, 23]. The transmitter sends multimedia contents at assigned time slot, while the receiver is listening, so they must be synchronous. Clock synchronization is critical in TDMA mechanism [24]. All nodes in A-LNT are synchronized when they join the network. However, the clock error will increase over time, which is caused by crystal tolerance and MCU clock accuracy. The clock error would lead to radio channel conflict, transmission failure, and system error. So periodic clock synchronization is necessary to maintain that the WASN works normally. The synchronization frequency should be a tradeoff decided by audio processing period and clock error. For a low-cost ±20 ppm crystal, in the worst case, the time error will reach 2 ms in 50 seconds; for a ±5 ppm crystal, the time will be 200 s, which is about 3 minutes.

The superframe time is decided by the audio sampling period and wireless period. In A-LNT, the codec bit rate is 15.625 Kbps; the audio codec generates bit stream continuous to MCU and demands the same number of bits from MCU in voice communications. In order to reduce complexity and power consumption, encoded audio content bytes generated by the codec should be less than the TX buffer size, which is 64 bytes. We design the superframe period T as 20.48 ms, which is also the audio sampling period; 40 bytes are sent to MCU in one T. In programming, we simplify operation by sending data to the codec; when receiving data from the codec, no additional timer or synchronization is required in this way. The audio data processing is mainly finished in the SPI interrupt function; in the interrupt function, MCU sends 1 byte decoded audio content when it receives 1 byte encoded audio from the audio codec.

ANODE sends encoded audio data periodically and receives wireless audio data from another ANODE in voice communications. In order to reduce hardware requirement and guarantee communication quality, we introduce four data buffers for cross access; two buffers are for storing received audio data, and the other two buffers are for storing encoded audio data. The maximum voice time delay is less than 2 times of T.

The superframe is divided into several time slots for sensing data transmission, network management packet transmission, and audio data transmission. The number of time slots is decided by the packet processing time. In order to get precise packet processing time, the packet processing time model is introduced as:

$$T_{\text{send}} = T_s + T_w + \frac{N_p + N_a}{B},$$
$$T_{\text{rev}} = T_r + T_m + \frac{N_p + N_a}{B}, \quad (1)$$

where T_{send} is the sending packet processing time; T_{rev} is the receiving packet processing time; T_s is the transmitter MCU processing time; T_w is the radio sending time; T_m is the receiver MCU processing time; T_r is the wireless receiving time; N_p is the packet payload length in bits; N_a is the preamble bits, sync word, and other data inserted automatically by CC2500; and B is radio transmission speed.

The packet processing time is decided by MCU main clock, data transmitting speed, packet length, and radio transmission speed. During voice communications, MCU main clock is 16 MHz, SPI speed is 4 Mbps, audio packet length is 46 bytes, and wireless speed is 500 kbps. T_{send} is 1.95 ms, which includes synthesizer calibration time 721 us. So the time slot for voice communication should be longer than 1.95 ms, and the time slot for network management and data transmission should be much longer than audio time slots for two reasons: at first, acknowledgment is usually necessary

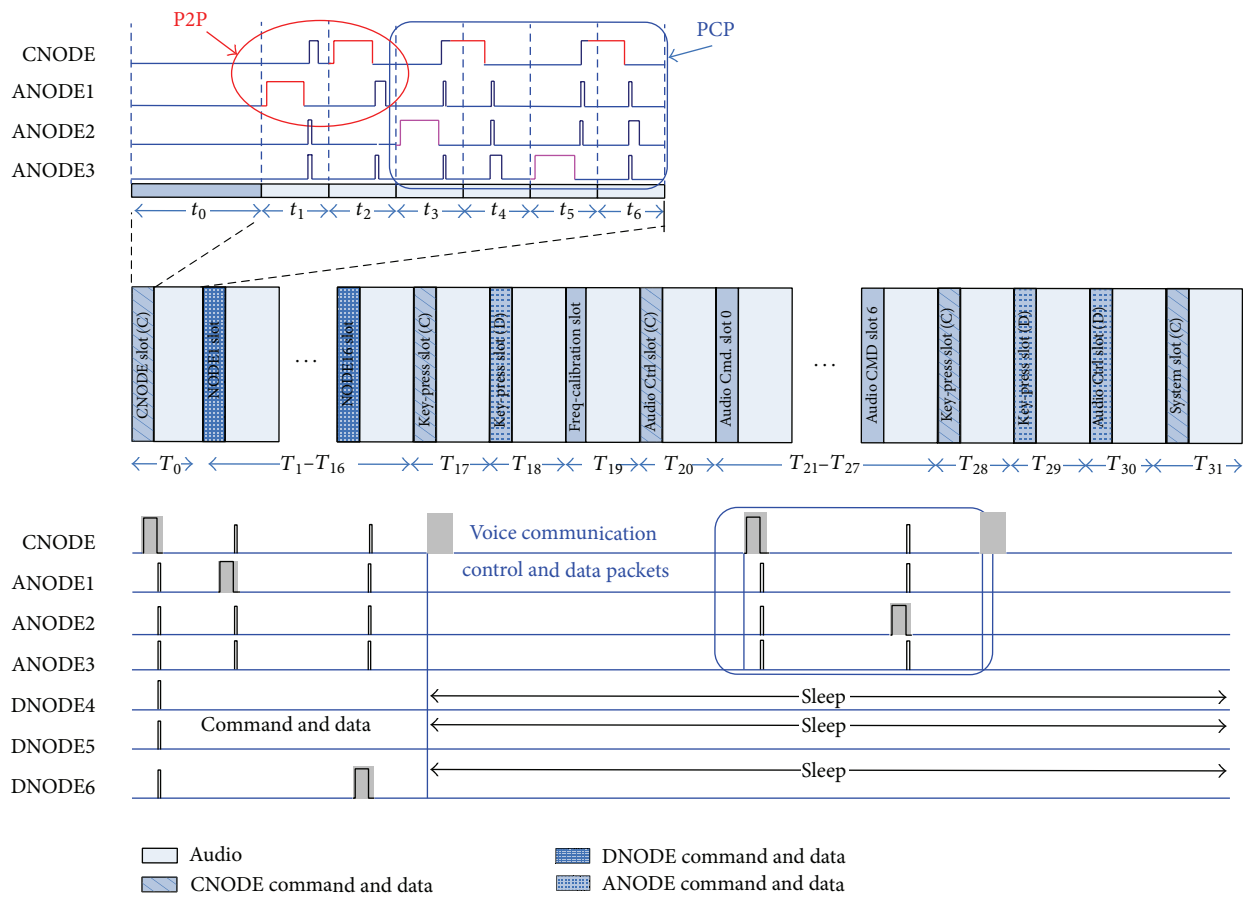

Figure 4: A-LNT channel allocation and address filtering diagram.

in network managements and it needs about twice the time; the other reason is that when the network works on low-speed mode, nodes do not have precise high-speed crystal or main clock; they need longer time for safety time interval and packet processing.

In multimedia communication applications, multimedia sensor nodes should be clock synchronized precisely with each other. However, high precision requires higher MCU clock, more expensive hardware, more current consumption, and shorter battery lifetime. In order to reduce power consumption, the audio processing units are shut down after voice communication finished and a superframe-based hybrid MAC protocol is introduced.

This MAC protocol mechanism consists of 4 key components: (1) the superframe is derived into data subframe and voice subframes. DNODEs listen to radio channels and send data only in data subframe; voice communications are carried out only in voice subframes. (2) The network adopts low time synchronization accuracy and lower node MCLK (main clock) to reduce energy consumption. The data subframe times are automaticly adjusted with network loads and CSMA/CA mechanism is adopted to manage radio channels. (3) When there are audio data transmissions, node MCLK is increased to work in full-speed mode and adopts high-precise time synchronization to ensure network performance. (4) Radio channels are allocated by center node using TDMA mechanism in voice subframes.

In detail, A-LNT consists of 1 CNODE, up to 16 ANODEs, and 64 DNODEs and an optional computer. The superframe T is 20.48 ms, and high main frequency is 16 MHz; the superframe is divided into 1 data subframe (t_0 = 6.08 ms) and 6 voice subframes (t_1–t_6 = 2.4 ms). When there are voice communications, ANODEs and CNODE listen to radio channels and send encoded audio content at specified voice subframes, and all DNODEs are in sleep. This platform supports up to 3-way P2P voice communications or 1-way PCP voice communication or a VCF including CNODE and all ANODES; the typical voice delay is less than 10 ms, and the time delay is less than 40 ms in the worst case. 32T compose a management cycle TT, the data subframe in the first superframe T_0 is a CNODE slot for network management. The data subframes in T_1–T_{16} are DNODE slots for network managements, network heart-beating, and sensing data transmissions. The data subframes in T_{17}–T_{31} are voice communication control and management slots. Every 6.55 s (i.e., 320T, the worst error at 20 ppm is 256 μs), CNODE broadcasts a polling packet POLL; all nodes should receive the packet, get time information, and adjust time to be in agreement with CNODE and then send reply packet ACK_POLL in specified slot, if there are sensor data need to

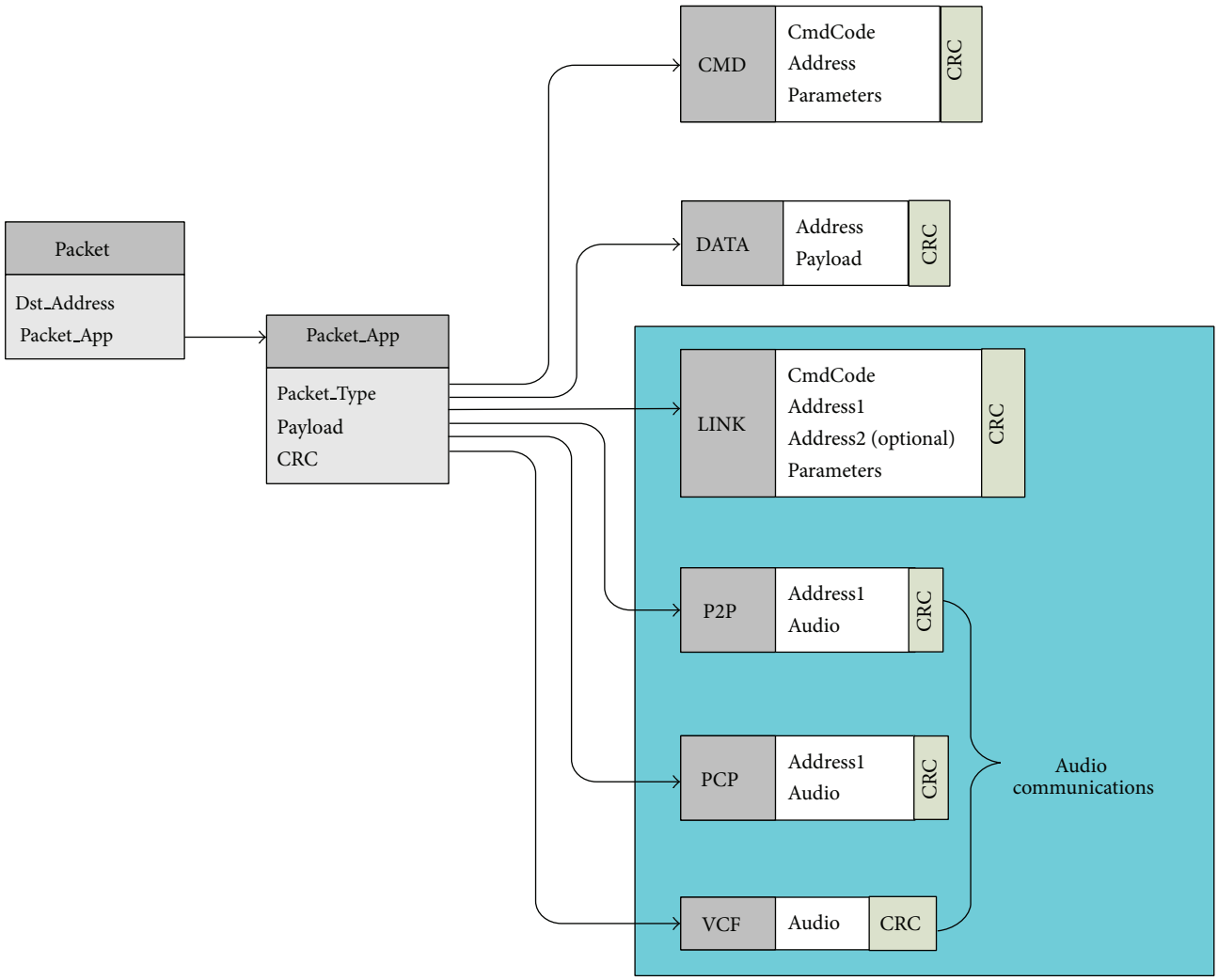

Figure 5: Packets structures.

be upload the nodes send them simultaneously by sending packet ACK_POLL_DATA instead of ACK_POLL; CNODE should send reply packet DATA_REVED to the node after receiving the packet immediately. For example, ANODE sends POLL at TT0. Nodes number 1–16 should send reply ACK_POLL/ACK_POLL_DATA in TT0, nodes number 17–32 should send reply ACK_POLL/ACK_POLL_DATA in TT1, and all replies should be finished in TT3. If CNODE did not receive ACK_POLL/ACK_POLL_DATA from one node for three successive management cycles, CNODE would delete the node. Priority design rules are as follows: CNODE has the highest priority, DNODEs have the highest priority in allocated slot, and other data are sent sequentially according to the priority within the data subframe. The high-speed crystal is shutdown when there is no voice communication, the node MCLK drops to about 2 MHz, and TT increases to 65.5 s (i.e., 3200 T; the worst error at 20 ppm is 2.56 ms). All nodes wake up in T_0 when the cycle is CNODE spooling cycle and go to sleep until it is time to send reply packet. When new node appears, it applies channel through CSMA/CA mechanism.

Address filtering is applied in A-LNT to reduce system total power consumption. In wireless network, all active nodes listen to radio channels; in most cases, only one node is the target node; other nodes receive useless packet and waste time to unpack and handle it. Address filtering is introduced to reduce wireless data processing time, which means that the wireless packet is unpacked and handled when address is matched; otherwise, the packet is abandoned. Address filtering can reduce processing time of that complete reception. This adaptive hybrid channel allocation method is an effective solution to the contradiction between multimedia communications and system power consumption. Figure 4 is the basic scheme of A-LNT channel allocation and address filtering protocol.

The other important pieces of information about A-LNT are as following.

The address allocation rule in the design is the following:

0X00: broadcasting address;

0x01: CNODE address;

0x02-0x0F: ANODE address;

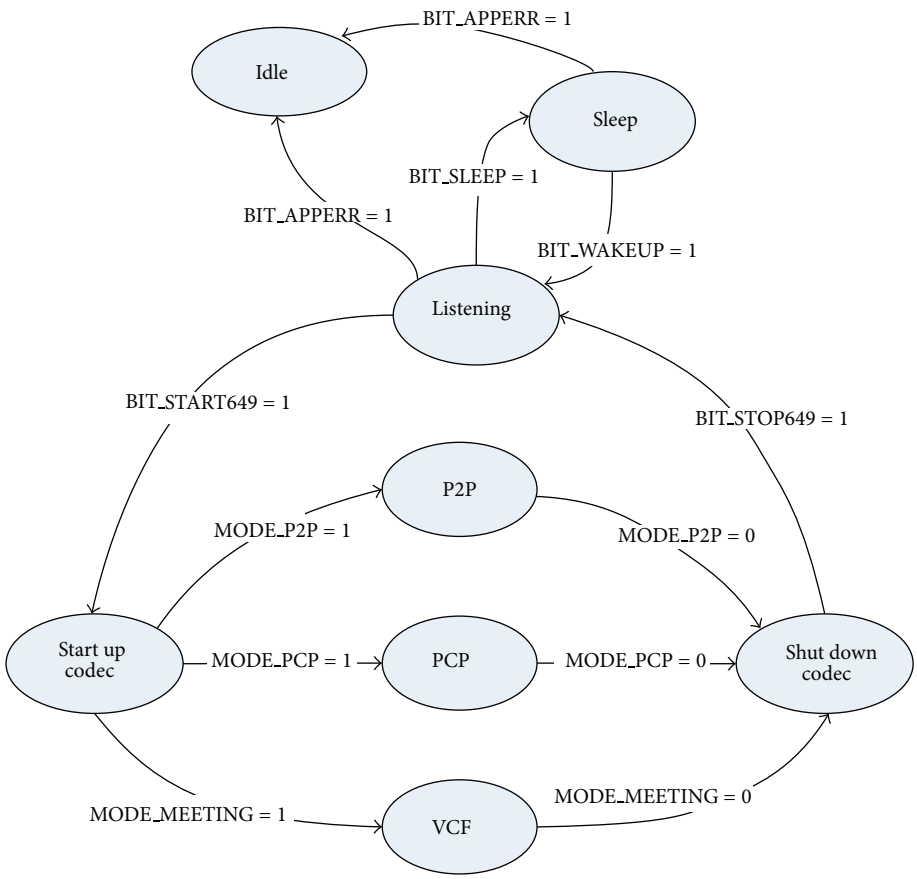

Figure 6: State machine of ANODE.

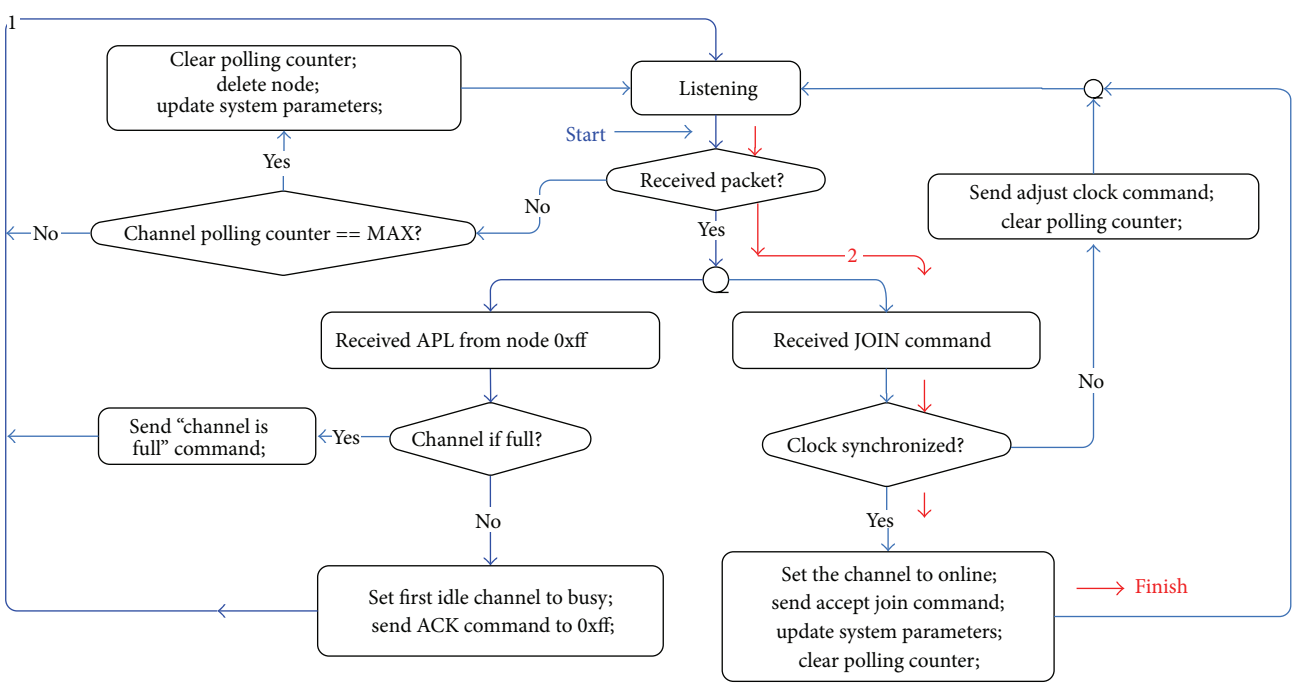

Figure 7: Simplified flowchart of node joining-CNODE part.

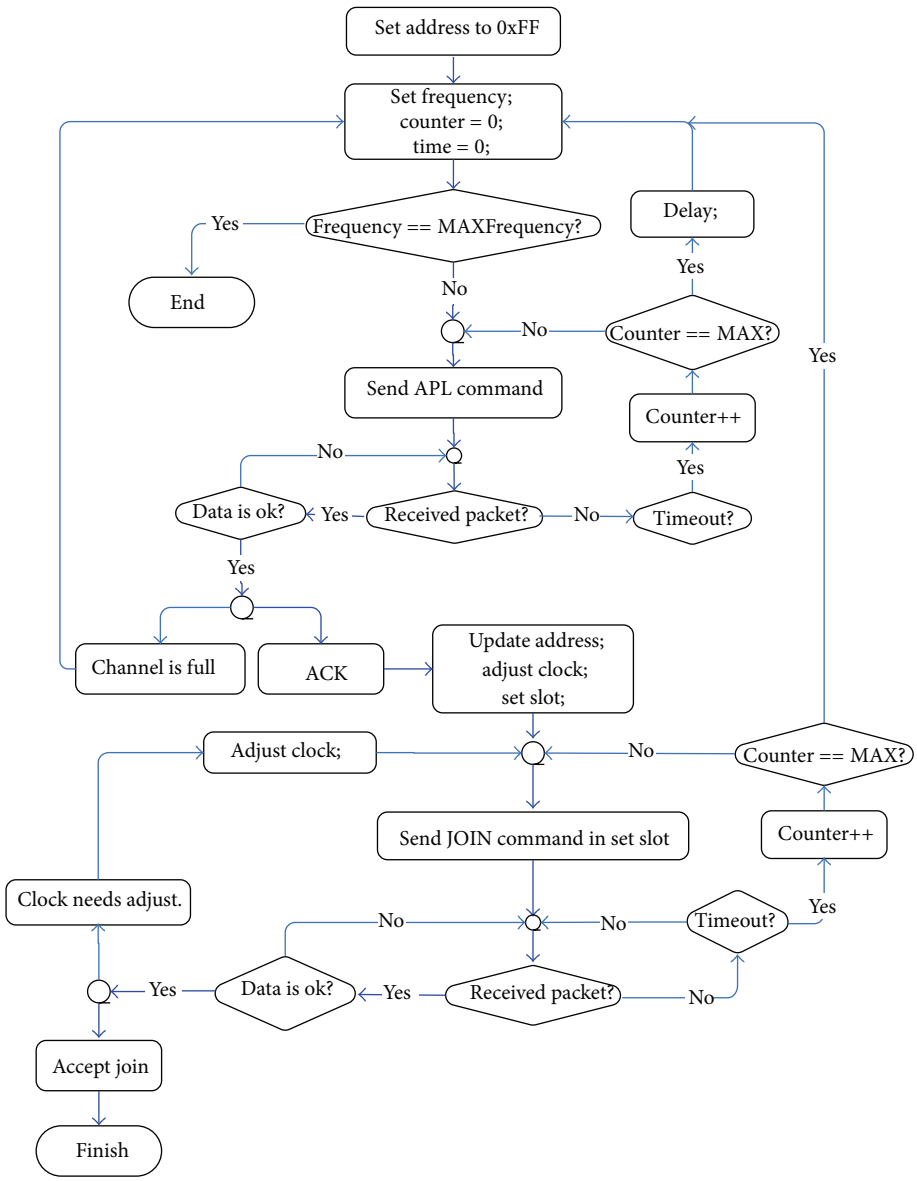

Figure 8: Simplified flowchart of node joining-ANODE/DNODE part.

0x10-0x7F: DNODE address;

0x80-0xEF: unused;

0xF0-0xFF: temporary address;

the wireless packets are divided into 6 types:

CMD, network management, priority 1;

DATA, sensing data and other data transmissions, priority 2;

LINK, voice communication link management and communication management, priority 3;

P2P, peer-to-peer audio data, priority 4;

PCP, peer-central-peer audio data, priority 5;

VCF, voice conference audio data, priority 6.

For more information about packet types, refer to Figure 5.

The state machine of ANODE is shown in Figure 6. ANODEs are sleeping in most times and go to listening mode at specified times. Listening mode could go to three voice communication modes, and audio codec only works in voice communication modes.

4.3. Network Management Protocol Realization. The last part of MAC protocol design is network management. The network management protocol takes on the role of the following:

(1) clock synchronization,

(2) radio channel management,

(3) nodes management.

```
Set RF channel = CHANNEL0
 while (1)
    Check RF channel
    if RF channel is not available
        If RF channel == MAX_RF_CHANNEL
            Halt application.
        else
            RF channel +1;
            continue;
    else
        break;
    while (1)
        Sleep;
        Wake up for RF packet receiving or system event
        if time == T_SENDING_SYNC // time for sending synchronization
            Send POLL(ALL_NODES);
        else if received RF packet
            Unpack the packet, extract source address NODEA;
            If packet type == ACK_POLL
                if the node is synchronized
                    TOUT_NODE = 0; // clear timeout counter of the node
            else if packet type == ACK_POLL_DATA
                if the node is synchronized
                    TOUT_NODE = 0;
                    Send DATA_REVED(NODEA);
            else if packet type == APPLY
                if node-list is not full
                    Assign a node number;
                    Update node-list;
                    Send ACK_APL(NODEA);
                else
                    Send DENY(NODEA);
            else if packet type == JOIN
                if node is synchronized
                    Update node-list; // node has joined the network by now.
                    Send ACP_JOIN(NODEA)
                else
                    Send ACK_ JOIN (NODEA);
            else if packet type == QueryPCP or QueryP2P // voice communication inquiry
                extract distinct address NODEB
                If free audio channel >0
                    if (NODEB == CNODE)
                        Send ACPPCP(NODEA);
                        Start codec;
                        Allocate audio channel;
                        Start voice communication;
                    else if. . . // more voice communication details, please browse the appendix.
                        . . .;
                else
                    Send DENY(NODEA);
            else if packet type == Audio data
                . . .Packet processing; // please browse the appendix.
            else if KEY pressed
                Response to KEY press;
            else if time == T_CHECK_NODE // check nodes status
                for each node in node-list
                    If TOUT_NODE > TOUTMAX
                        Delete the node;
            else
                . . .;
```

Algorithm 1

```
(1)  Set address = 0xFF
(2)  Set RF channel = CHANNEL0
(2)  while (1)
(3)      Set counter = 0;
(4)      Listen to radio channel for 100 ms
(5)      if RF packet received
(6)          Calculate system time
(7)      Set system parameter;
(8)      while (counter < CMAX)
(9)          Send APPLY(CNODE)
(10)         while (t < TTIMEOUT)
(11)             Listen to radio channel
(12)                 if ACK_APL received
(13)                     break;
(14)                 else if DENY received
(15)                         break;
(16)         if (t ≥ TTIMEOUT)
(17)             counter++;
(18)         else if DENY received
(19)             RF channel = unavailable;
(20)             break;
(21)         else if ACK_APL received
(22)             Jcounter = 0;
(23)             While (Jcounter < JMAX)
(24)                 Synchronize clock;
(25)                 Change address to new address
(26)                 Send JOIN(CNODE) at assigned time slot
(27)                 while (t < TTIMEOUT)
(28)                     Listen to radio channel
(29)                     if ACP_JOIN received
(30)                         break;
(31)                     else if ACK_JOIN received
(32)                         Synchronize clock;
(33)                         Send JOIN(CNODE) at assigned time slot
(34)                 if (network joined)
(35)                     Break;
(36)                 if (t ≥ TTIMEOUT)
(37)                     Jcounter++;
(38)     if (network joined)
(39)         Update system parameters;
(40)         break;
(41)     else if (RF channel is unavailable)
(42)         If RF channel == MAX_RF_CHANNEL
(43)             Halt application.
(44)         else
(45)             RF channel++;
(46)             continue;
(47) while (1)
(48)     Sleep;
(49)     Wake up for RF packet receiving or system event
(50)     if time == T_ACK_POLL // time for sending ack-synchronization
(51)         if data needs to be sent
(52)             Send ACK_POLL_DATA(CNODE);
(53)         else
(54)             Send ACK_POLL (CNODE);
(55)     else if received RF packet
(56)         Unpack the packet, extract source address NODEA;
(57)         if packet type == POLL
(58)             Synchronize clock;
(59)             TOUT_NODE = 0; //clear timeout counter of the node
```

Algorithm 2: Continued.

```
else if packet type == DATA_REVED
    Clear sent data information;
else if packet type == QueryP2P // voice communication inquiry
    ...; // more voice communication details please browse the appendix.
else if packet type == audio data
    Packet processing; // please browse the appendix.
else if KEY pressed
    Response to KEY press;
else if time == T_CHECK_NODE // CNODE status
    If TOUT_NODE > TOUTMAX
        reset application;
else...
    ...;
```

Algorithm 2

In our design, the work is mainly done by CNODE. It chooses an available radio channel and waits for radio packet receiving events. When a packet is received, CNODE does corresponding operation according to packet type. The network management process pseudocode is as shown in Algorithm 1.

The simplified flowchart of node joining is shown in Figure 7.

In order to join A-LNT, wireless node should seek an active CNODE, send APPLY to CNODE, get time information, and synchronize with CNODE. The process is as shown in Algorithm 2.

The simplified flowchart of node joining is shown in Figure 8.

By now, the A-LNT MAC protocol design is finished; it is simple and efficient, it consumes limited RAM and flash resources, and the details information is the following:

CNODE: 17 KB ROM, 0.5 KB RAM;

ANODE: 11 KB ROM, 0.4 KB RAM;

DNODE: 2.6 KB ROM, 0.1 KB RAM.

We would not discuss voice communication details here and have attached it to the end of the paper as it is tedious. In the next section, we will carry out experiments to verify A-LNT platform performance and discuss the results.

5. Results and Conclusion

At first, we measured the operating currents of three types of nodes (Table 1). The current consumption is mainly determined by audio unit and radio unit. It could be lower through reducing output power and receiving sensitivity of CC2500, turning the volume down of earphone, and powering down the LCM.

Then, we have studied the batteries lifetime in theory. In order to simplify calculation, we assume that the battery maintains OCV (constant open circuit voltage) 1.5 V and RI (the internal resistance) 150 mΩ. Battery capacity Q is 2300 mAh. The batteries are three alkaline batteries in series. The ANODE currents vary with operation mode: TX, RX, sleep, and audio. Average TX time tTX is 2 ms every 1000 T; the RX time tRX is 3 ms every 1000 T. For boards with only LDO, the battery lifetime is given as follows:

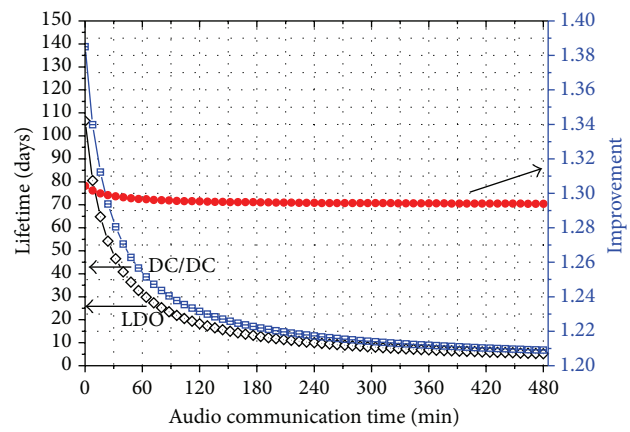

Figure 9: Battery lifetime versus voice communication time.

$$T_{\text{Days}} = (Q \times 3600) \times \left(I\text{audio} \times t\text{audio} + I\text{rx} \times t\text{RX} + I\text{tx} \times t\text{TX} + I\text{sleep} \times t\text{sleep} + 3600 \times 24 \times I\text{ldo}\right)^{-1}, \quad (2)$$

where Ildo is supply current of LDO; in the design the LDO is XC6204B30 and Ildo is 70 uA.

For boards with DC/DC converters, battery lifetime is given as follows:

$$T_{\text{Days}} = (Q \times 3600) \times \left(I\text{audio}' \times t\text{audio} + I\text{rx}' \times t\text{RX} + I\text{tx}' \times t\text{TX} + I\text{sleep}' \times t\text{sleep} + 3600 \times 24 \times I\text{ldo}'\right)^{-1}, \quad (3)$$

where

$$\text{Ii}' = \frac{\text{Ii} \times V_{\text{out}}}{(\text{OCV} - \text{RI} \times \text{Ii}') \times \eta} \quad (4)$$

and $V_{\text{out}} = 3.3$ V, $\eta = 95\%$.

Table 1: Current summary.

Node type	TX (mA)	RX (mA)	Audio (mA)	Average current (mA)	Sleep mode (mA)
CNODE	21.4	19.6	27	53	3.2
ANODE	21.4	19.6	27	52	0.9
DNODE	21.4	19.6	—	22	0.6

Table 2: Processing times of different packet lengths and hardware.

Time (us)	Address match	MCLK (MHz)	SPI speed (Kbps)	Payload length (bytes)
200	Yes	16	4000	5
45	No	16	4000	5
900	Yes	2	500	5
160	No	2	500	5
350	Yes	16	4000	12
45	No	16	4000	12
1500	Yes	2	500	12
160	No	2	500	12
1050	Yes	16	4000	46
45	No	16	4000	46

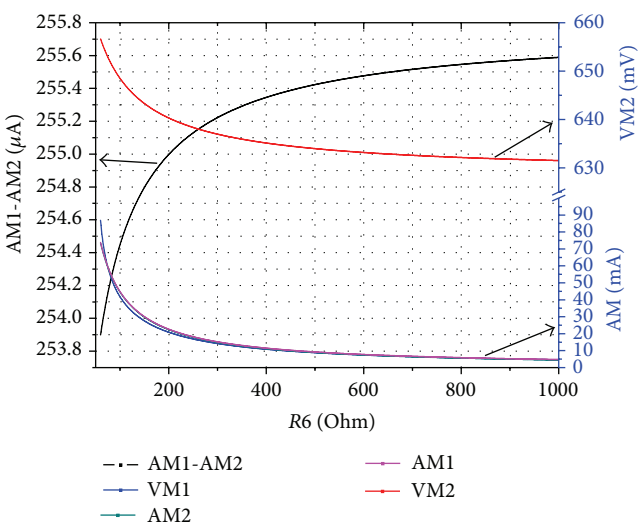

Figure 10: Voltages and currents versus loads.

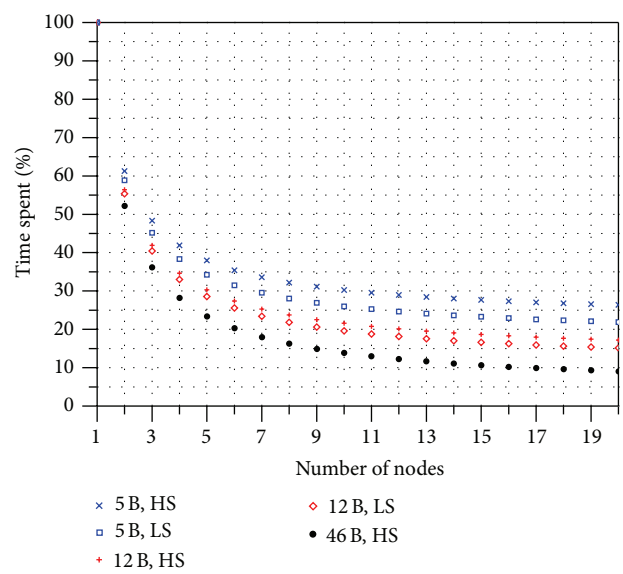

Figure 11: Time spent versus number of nodes.

Figure 9 shows the calculation results. The DC/DC converter extends battery lifetime by greater than 29%. If voice communication time is 30 minutes per day, ANODEs could work for more than 60 days without changing batteries. It is also possible to serialize more batteries or use high voltage batteries extending node working time.

The minimum input voltage is calculated by the following equations:

$$V_{\text{in min}} = V_{\text{out max}} + \text{IL}_{\max} \times \left(r_{\text{ds}}\,(\text{ON})\max + \text{RL}\right),$$
$$\text{IL}_{\max} = I_{\text{out max}} + \frac{V_{\text{out}} \times \left(1 - \left(V_{\text{out}}/V_{\text{in}}\right)\right) / \left(L \times f\right)}{2}, \quad (5)$$

where $I_{\text{out max}} = 53\,\text{mA}$; $V_{\text{out}} = 3.3\,\text{V}$; $L = 10\,\mu\text{H}$; $f = 1\,\text{MHz}$; $r_{\text{ds}}(\text{ON})\max = 670\,\text{m}\Omega$; the power inductance is CDRH5D28NP-100N from Sumida, and RL = 65 mΩ. The result shows that $V_{\text{in min}}$ is less than 3.4 V, for three alkaline batteries in series; the node could extract almost all energy.

The load switch circuit simulation is carried out by TINA-Ti 9.0; the results are shown in Figure 10. The current consumption of BISS transistor is about 255 uA, and the VBEsat is about 50 mV when the load current is 50 mA. Node shutdown current consumption is only 2.21 uA. The power management circuitry has virtually no impact on the node power consumption.

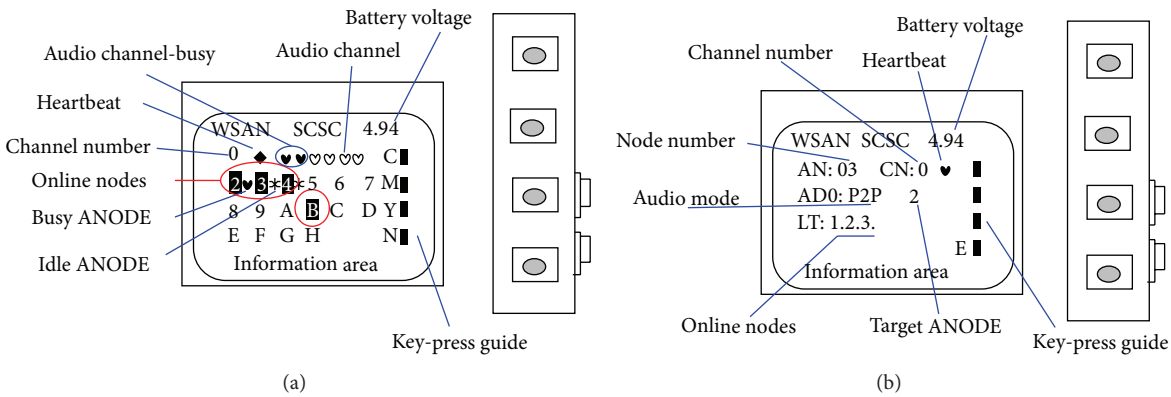

Figure 12: LCD screen of CNODE (a) and ANODE (b).

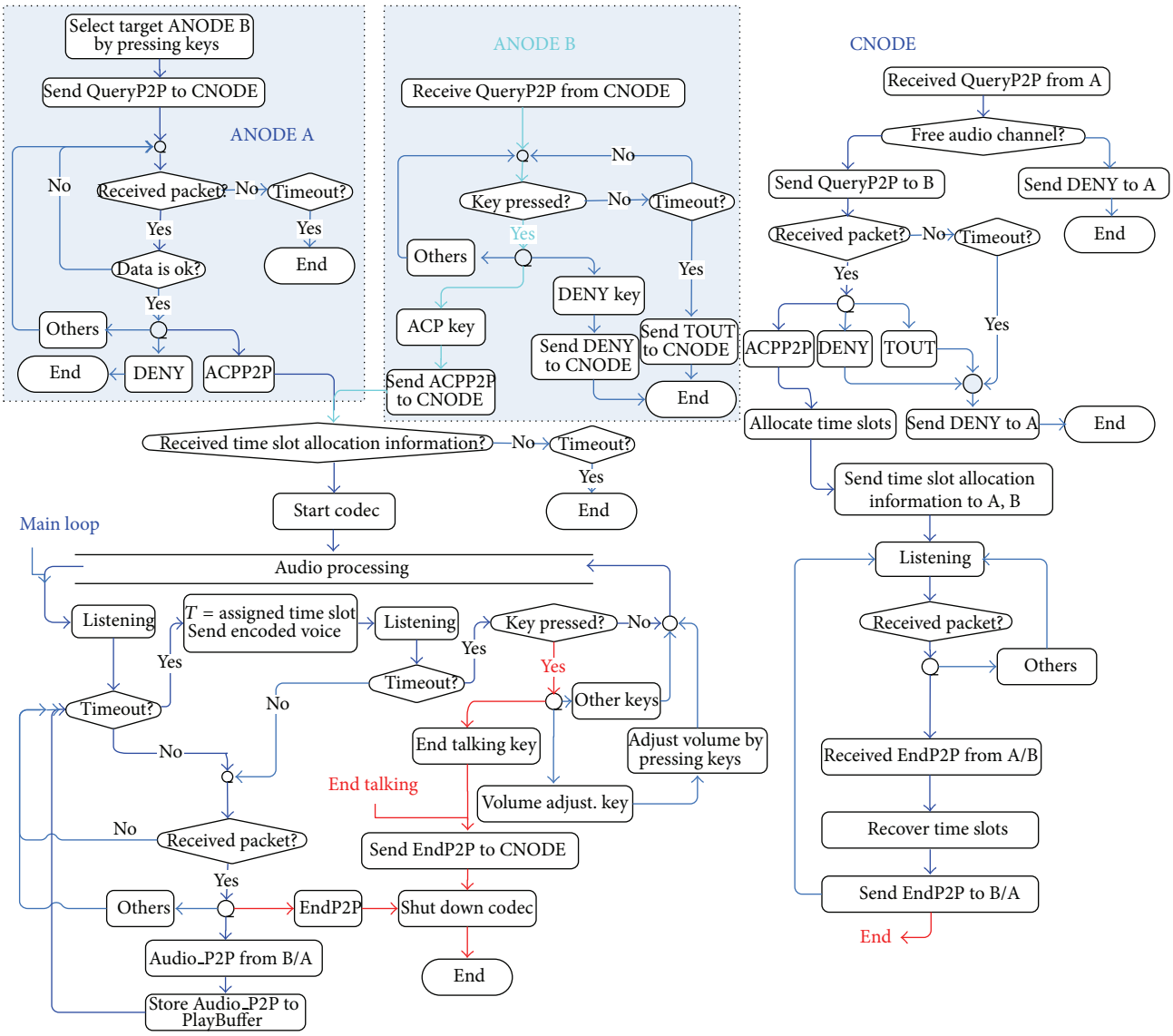

Figure 13: P2P communication flowchart.

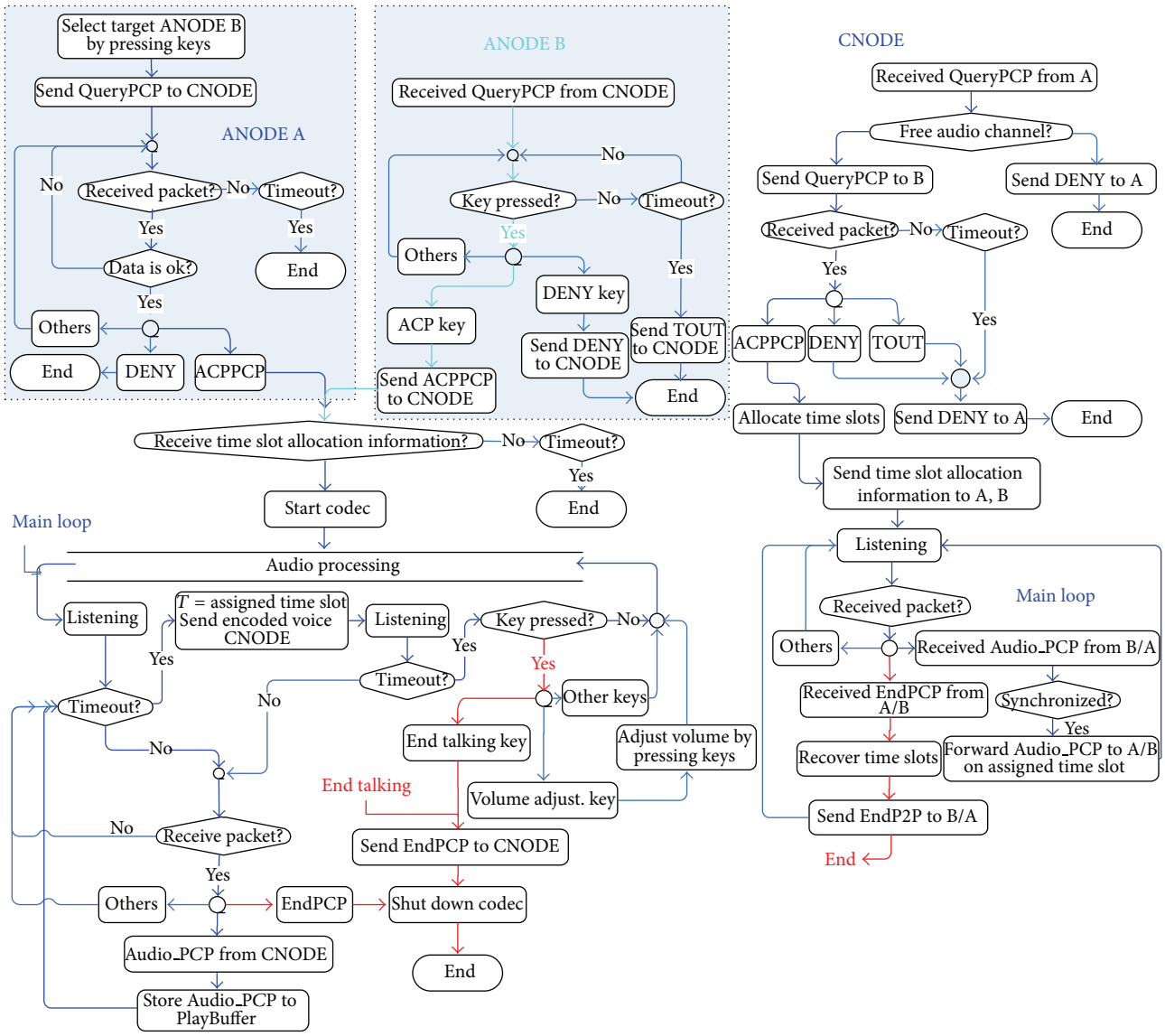

Figure 14: PCP communication flowchart.

The communication distance between ANODEs is longer than 70 meters indoors and 120 meters outdoors. The results are measured under the following conditions: line of sight, being about 1.5 meters above ground, and the nodes being placed on a table or carried by person.

At last, the address filtering performance is studied by measuring RF packet processing times. We measure packet processing times with a pair of nodes. Node A sends the same packets every 100 ms for 200 times. At the same time, node B stays in RX states when the RF packet is detected; the timer starts counting until the packet is received and processed, and then the timer count is stored into an array. At last, the processing time is calculated by averaging all 200 counts. The processing times of different packet lengths are measured as shown in Table 2.

The timing accuracy is 5 us in the above measures. The total processing time saving with parameters in Table 2 versus number of nodes is shown in Figure 11.

Where B means bytes, HS means "high speed" and LS means "low speed." It can be seen from Figure 11 that the processing time of all active nodes in WSN reduces with the node number increasing and packet length increasing. When the number of ANODEs reaches 6, the time spent is reduced to 20% of that without address filtering. It is an efficient method to reduce network power consumption.

In conclusion, we have presented a low-power WASN platform A-LNT from hardware realization to protocol design. The network has a star topology and three voice communication modes: P2P, PCP, and VCF. The audio codec is CVSD 15.625 kbps; cross accessing and data buffer pool

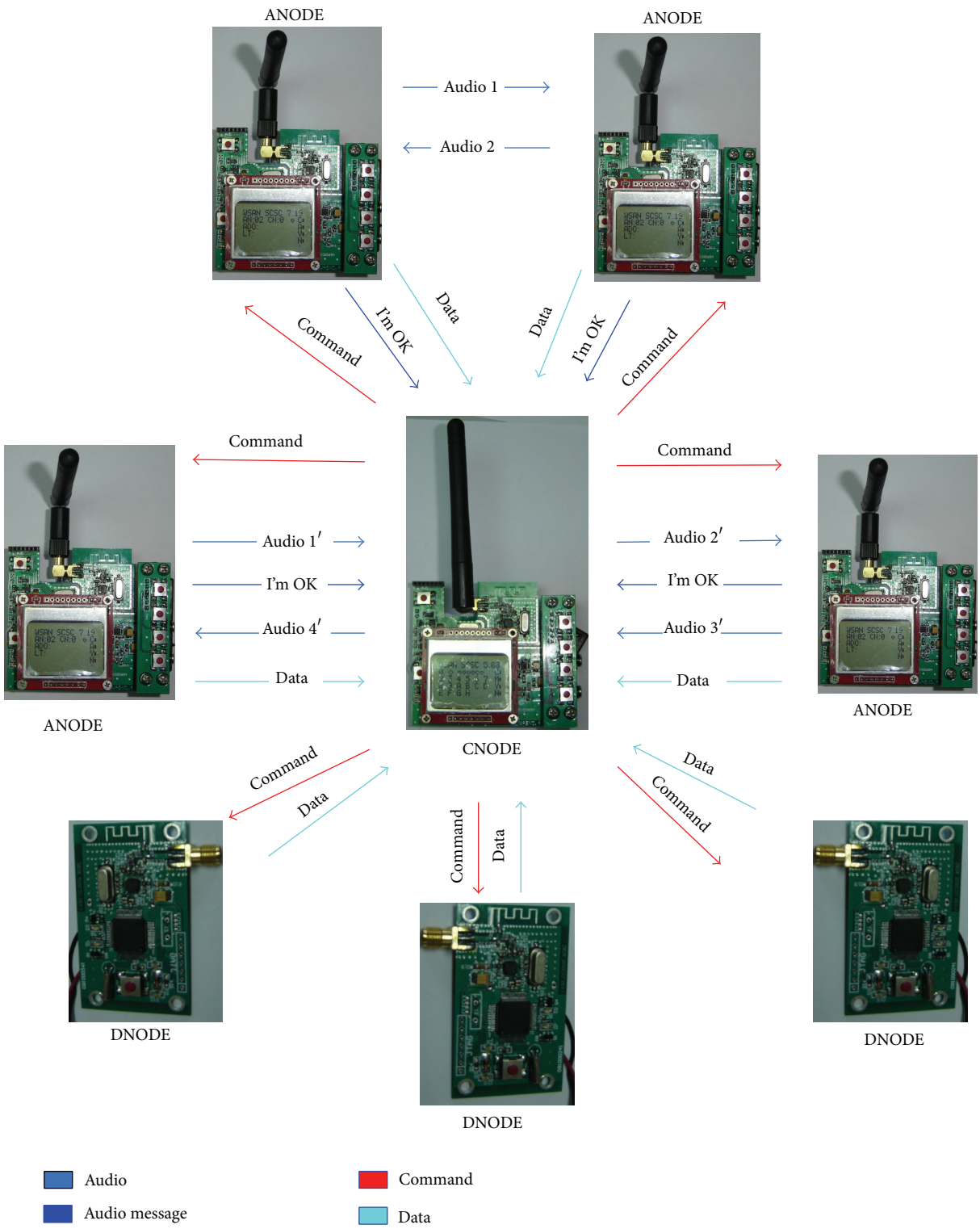

Figure 15: Schematic diagram of P2P and PCP.

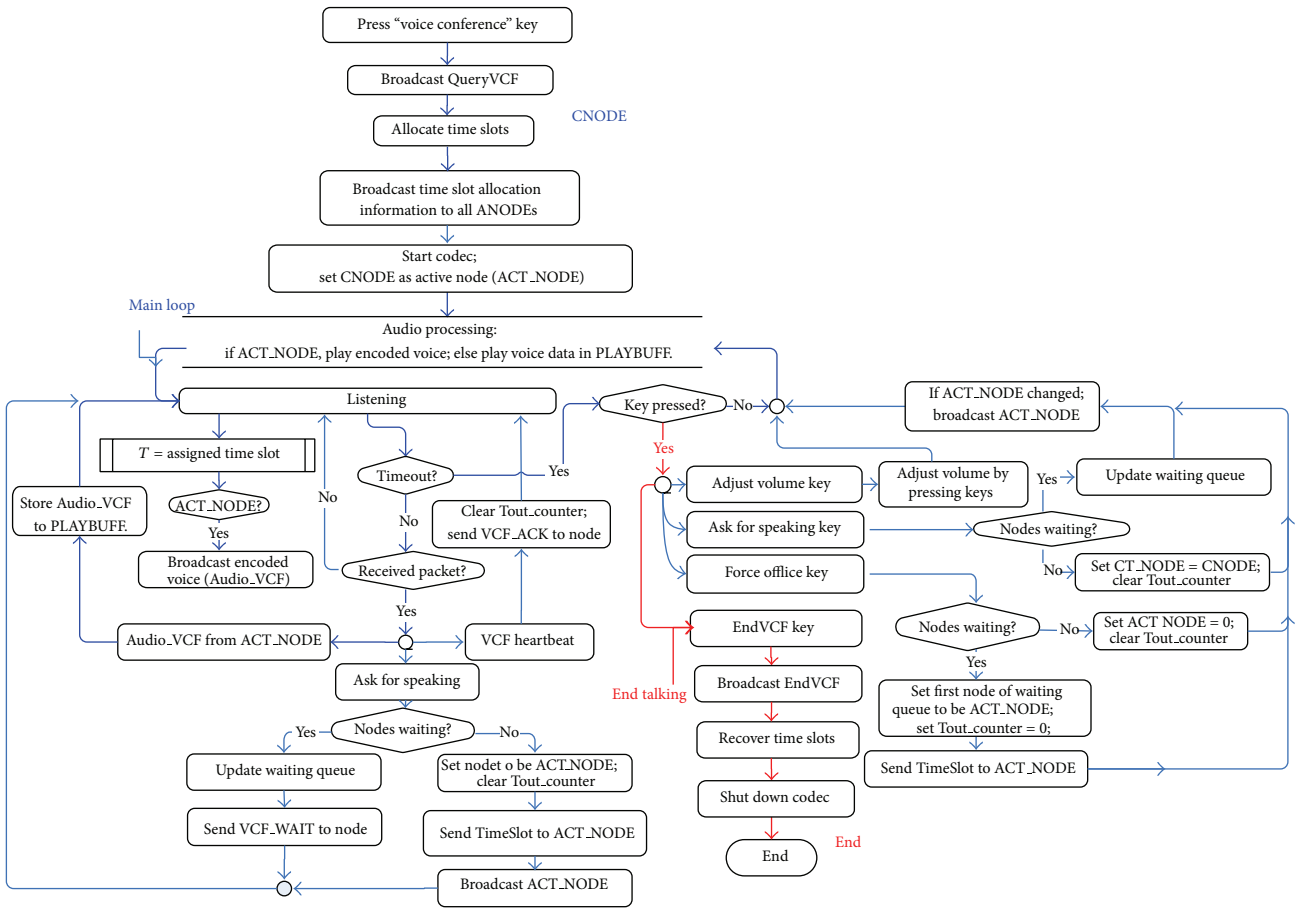

Figure 16: VCF communication flowchart—CNODE.

are introduced to reduce hardware requirement and power consumption. The efficient power management method for A-LNT is studied by simulation and mathematical theory in detail. We also designed a hybrid MAC protocol based on superframe. The superframe setting is studied by considering the audio sampling period, wireless packet processing, clock error, and voice communication channels. The superframe time is 20.48 ms; it is divided into a time slot (t_0, 6.08 ms) for network management and data transmission and 6 audio time slots (t_1–t_6, 2.4 ms). Address filtering and sleeping in specified time slots are applied to reduce wireless packet processing time and power consumption. MAC protocol key elements such as clock synchronization and network management are discussed also. The result shows that A-LNT is a low-power, low-speed, and high-performance WSN platform. It consists of up to 16 ANODEs, 64 DNODEs, and 1 CNODE. The audio channel capacity is 3 real-time two-way voice communications or audio conference including all audio nodes at the same time. And the voice delay is less than 40 ms. It suggests possible applications to emergency voice communication, audio/sound sensor network, health monitoring system, and so forth.

Our future work is concerned with increasing voice communication channel number and reducing power consumption. In detail, we are looking for new audio codec with low bit rate and low power, new type of wireless transceiver with fast switch speed from idle/sleep status to TX/RX status, high-speed and low-power wireless transceiver, and wireless protocol design.

Appendix

A. Voice Communication Process

There are three voice communication modes: P2P, PCP, and VCF. All communication managements are operated by CNODE. The voice communications are initiated and managed by pressing keys according to LCD screen guidance information (Figure 12). We will discuss these types of voice communications in detail.

A.1. P2P Mode. ANODE A selects target ANODE B by key pressing and sends "QueryP2P" command to CNODE; CNODE checks whether there are free audio time slots and whether ANODE B is free. If both conditions are met, CNODE forwards "QueryP2P" to ANODE B and waits for answering. If ANODE B denies the query or timeout, CNODE sends 'DENY' command to ANODE A and the

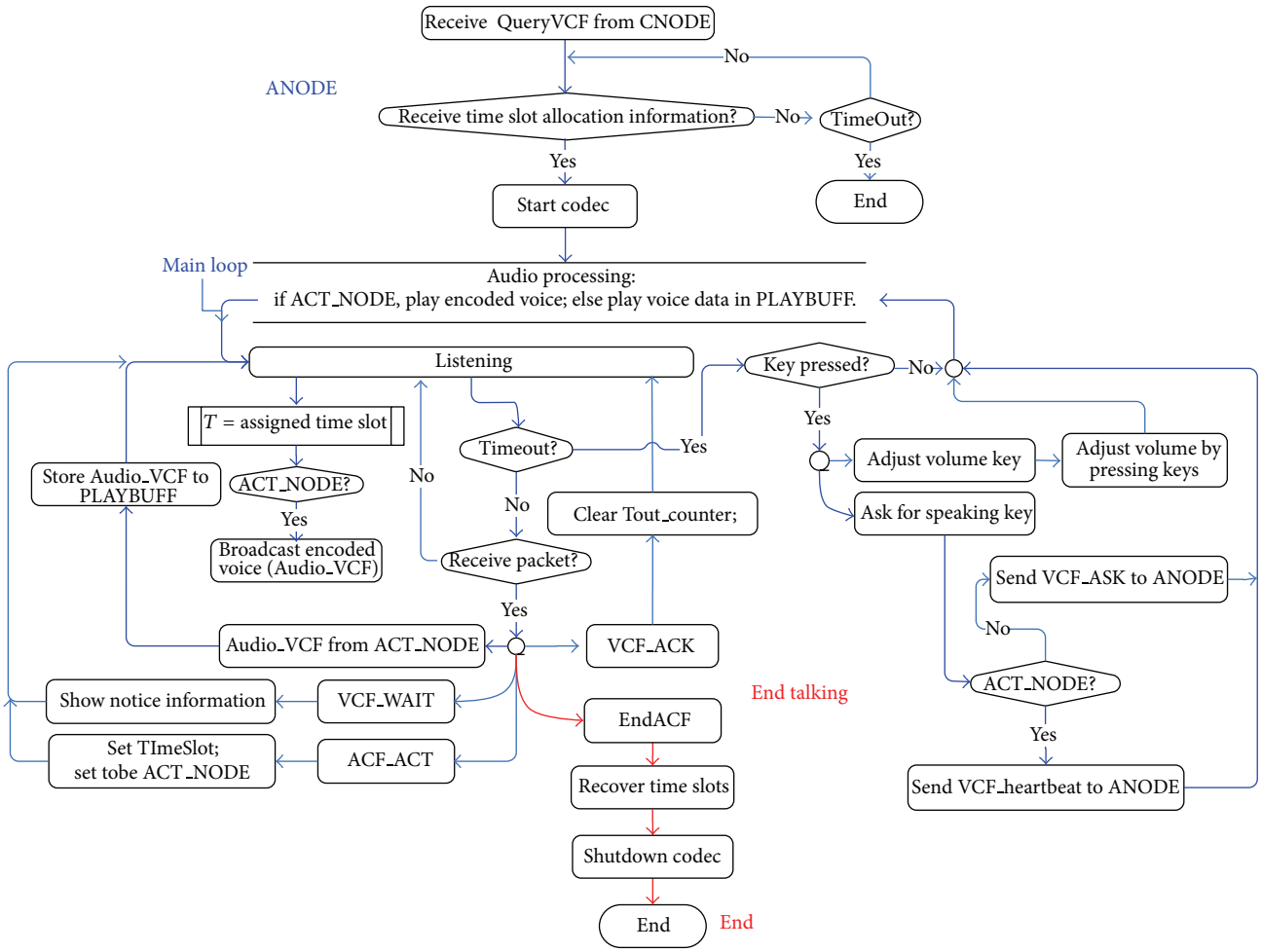

Figure 17: VCF communication flowchart—ANODE.

voice communication halts. If ANODE B accepts the query, CNODE allocates time slots to ANODEs A and B and checks communication status during voice communication. ANODEs A and B receive time slot information and then they start audio codec and send audio contents on assigned time slots. In P2P communication, ANODEs A and B send heartbeat packets to CNODE periodically. ANODE A/B could end communication by pressing "End communication" key. Meanwhile, the node sends "EndP2P" command to CNODE; CNODE withdraws the time slots and forwards the command to the other ANODE. The P2P communication is ended. The simplified flowchart is shown in Figure 13.

A.2. PCP Mode. PCP mode is similar to P2P mode; the differences are that CNODE allocates 4 time slots and all audio contents are forwarded by CNODE. The simplified flowchart is shown in Figure 14.

The schematic diagram of P2P and PCP is shown in Figure 15.

A.3. VCF Mode. VCF is initiated by CNODE. Any ANODE that wants to speak should send VCF_ASK command to CNODE; CNODE checks if there is an active speaker and sends respect reply to the asker. The VCF could be ended only by CNODE. The simplified flowchart is shown in Figures 16 and 17.

The schematic diagram of P2P and PCP is shown in Figure 18.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the Shandong Provincial Foundation for Outstanding Young Scientists (Grants nos. BS2012DX035 and BS2011DX031). The authors also thank the anonymous reviewers and the editor for their valuable comments.

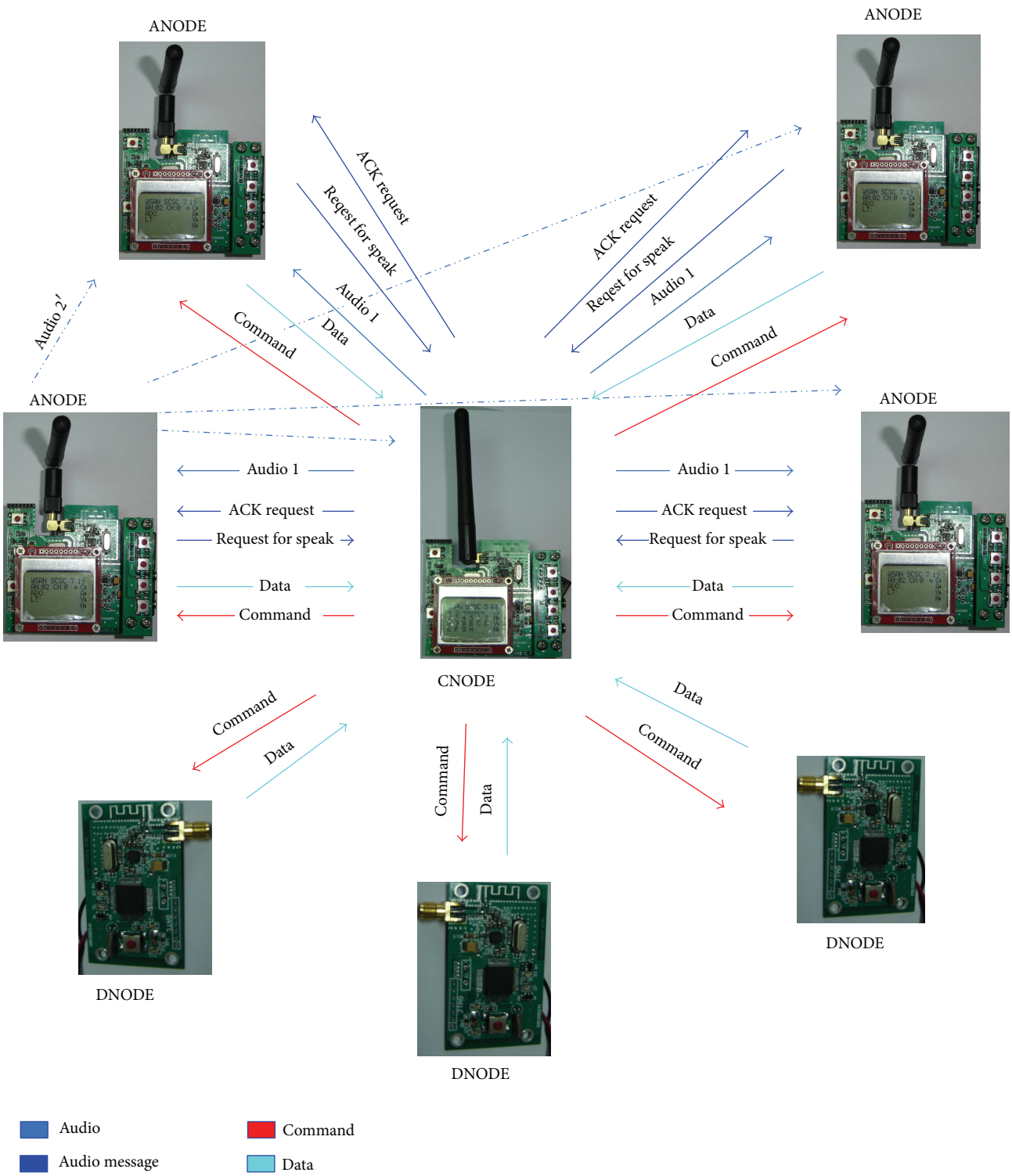

Figure 18: Schematic diagram of VCF.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] S. A. Khan, *Wireless Sensor Networks: Current Status and Future Trends*, CRC Press, New York, NY, USA, 2012.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[4] I. T. Almalkawi, M. G. Zapata, J. N. al-Karaki, and J. Morillo-Pozo, "Wireless multimedia sensor networks: current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662–6717, 2010.

[5] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, 2012.

[6] H. Touil, Y. Fakhri, and M. Benattou, "Energy-efficient MAC protocol based on IEEE 802. 11 e for Wireless Multimedia Sensor Networks," in *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS '12)*, pp. 53–58, 2012.

[7] Y. Zhenyu, L. Ming, and L. Wenjing, "CodePlay: live multimedia streaming in VANETs using symbol-level network coding," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 3006–3013, 2012.

[8] L. L. Hanzo, C. Somerville, and J. Woodard, *Voice and Audio Compression for Wireless Communications*, Wiley-IEEE Press, New York, NY, USA, 2008.

[9] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[10] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni, "LiT MAC: Addressing the challenges of effective voice communication in a low cost, low power wireless mesh network," in *Proceedings of the1st ACM Symposium on Computing for Development (DEV '10)*, p. 5, December 2010.

[11] Q. Li, M. Zhang, and G. Xu, "A novel element detection method in audio sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 607187, 12 pages, 2013.

[12] E. Touloupis, A. Meliones, and S. Apostolacos, "Speech codecs for high-quality voice over ZigBee applications: evaluation and implementation challenges," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 122–128, 2012.

[13] G. Zhao, H. Ma, Y. Sun, and H. Luo, "Design and implementation of enhanced surveillance platform with low-power wireless audio sensor network," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 854325, 18 pages, 2012.

[14] S. W. Jung, H. J. Lee, J. H. Lee, and S. H. Cho, "Interworking of Voice over Sensor Network (VoSN) using the TDMA/TDD MAC and VoIP based SIP," in *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC '12)*, pp. 97–101, 2012.

[15] V. Gabale, J. Patani, R. Mehta, R. Kalyanaraman, and B. Raman, "Building a low cost low power wireless network to enable voice communication in developing regions," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, pp. 2–15, 2012.

[16] J. M. Valin, "Speex: a free codec for free speech," in *Proceedings of the Australian National Linux Conference*, Dunedin, New Zealand, 2006.

[17] J. Kim, J. H. Yoo, J. H. Lee, and S. H. Cho, "A design of the full-duplex voice mixer for multi-user voice over sensor networks (VoSN) systems," in *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC '12)*, pp. 102–105, 2012.

[18] J. H. Lee, J. H. Yoo, and S. H. Cho, "A new TDMA/TDD MAC protocol design for voice communications based on the IEEE 802.15.4 PHY," in *Proceedings of the 27th International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 1–4, 2012.

[19] L. Herrera, A. Calveras, and M. Catalán, "A two-way radio communication across a multi-hop wireless sensor network based on a commercial IEEE 802.15. 4 compliant platform," *Procedia Engineering*, vol. 25, pp. 1045–1048, 2011.

[20] O. O. Khalifa, S. Khan, M. D. R. Islam, M. B. Muktar, and Z. Yaacob, "Speech coding for bluetooth with CVSD algorithm," in *Proceedings of the RF and Microwave Conference (RFM '04)*, pp. 227–229, October 2004.

[21] CML Microcircuits, "CMX649 Datasheet," http://pdf.datasheetcatalog.com/datasheets/480/101671_DS.pdf.

[22] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 1567–1576, June 2002.

[23] Y. Li, C. S. Chen, Y. Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey," in *Proceedings of the 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems (FeT '07)*, 2007.

[24] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.