



US 20250267054A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2025/0267054 A1

Motos et al.

(43) Pub. Date: Aug. 21, 2025

(54) OFDM COMMUNICATION WITH HOPPING SEQUENCE

(71) Applicant: TEXAS INSTRUMENTS INCORPORATED, Dallas, TX (US)

(72) Inventors: Tomas Motos, HAMAR (NO); Thomas Almholt, Murphy, TX (US)

(21) Appl. No.: 19/053,543

(22) Filed: Feb. 14, 2025

Related U.S. Application Data

(60) Provisional application No. 63/555,011, filed on Feb. 17, 2024.

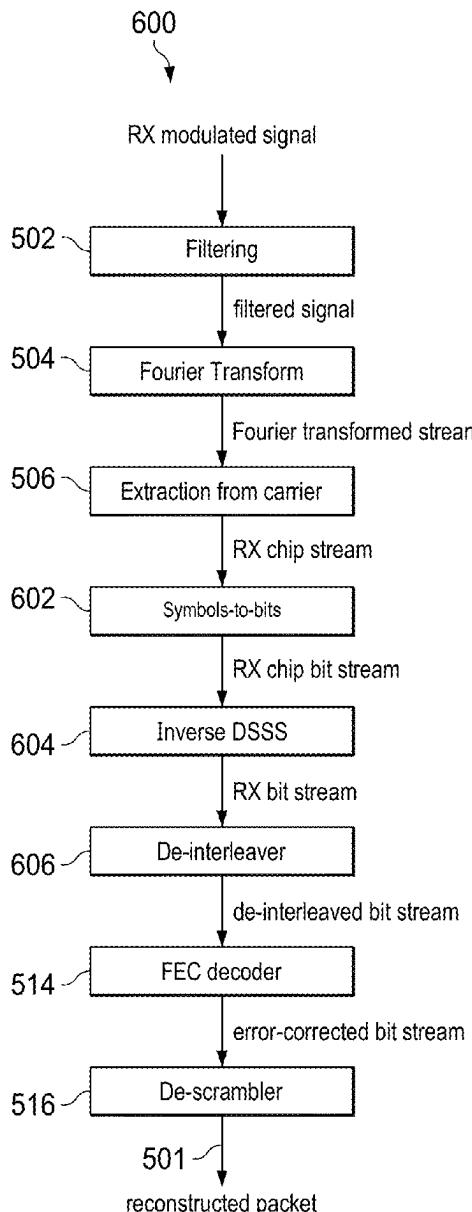
Publication Classification

(51) Int. Cl.
H04L 27/26 (2006.01)
H04L 5/00 (2006.01)

(52) U.S. Cl.
CPC *H04L 27/2673* (2013.01); *H04L 5/0012* (2013.01); *H04L 27/2605* (2013.01)

(57) ABSTRACT

In an embodiment, a method includes: transmitting, by a first device, a first synchronization sequence in a single synchronization channel of a plurality of channels; and after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.



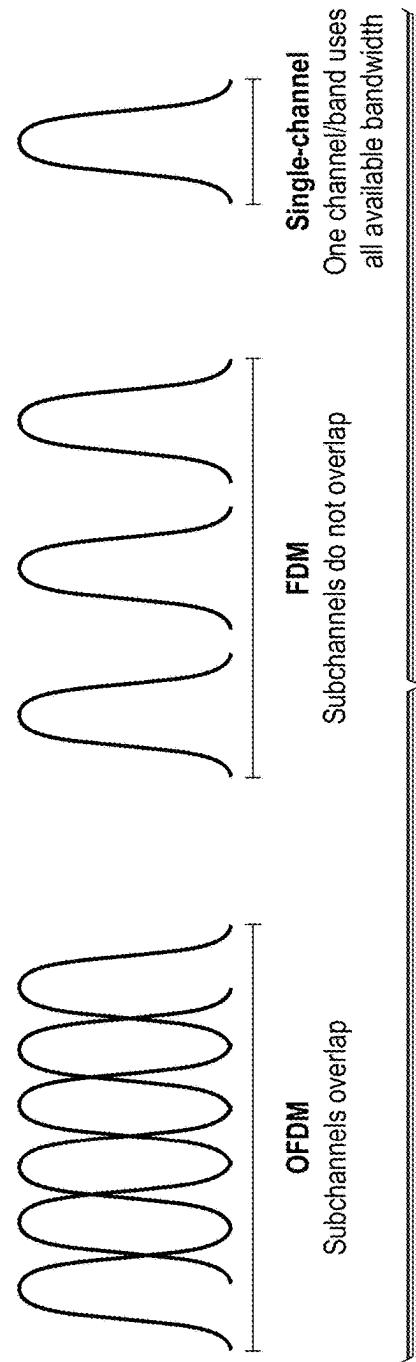


FIG. 1

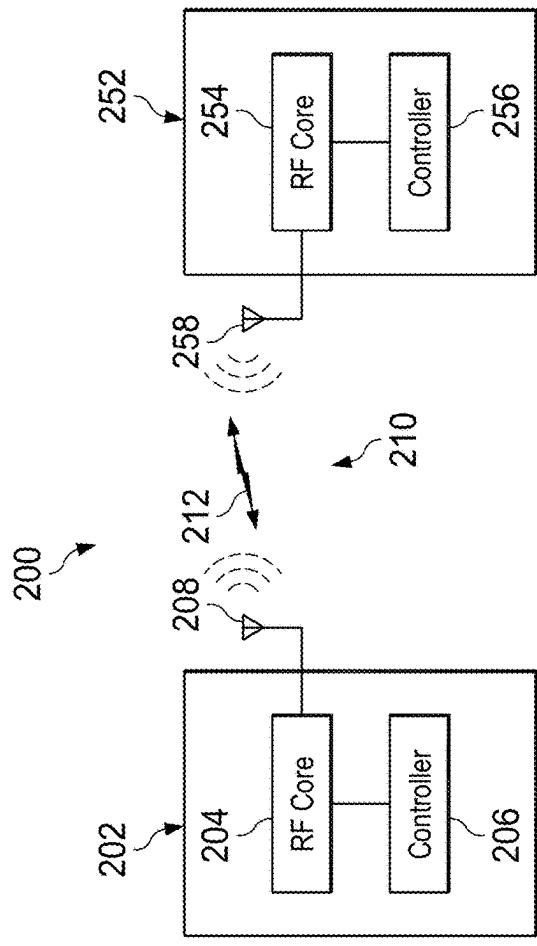


FIG. 2

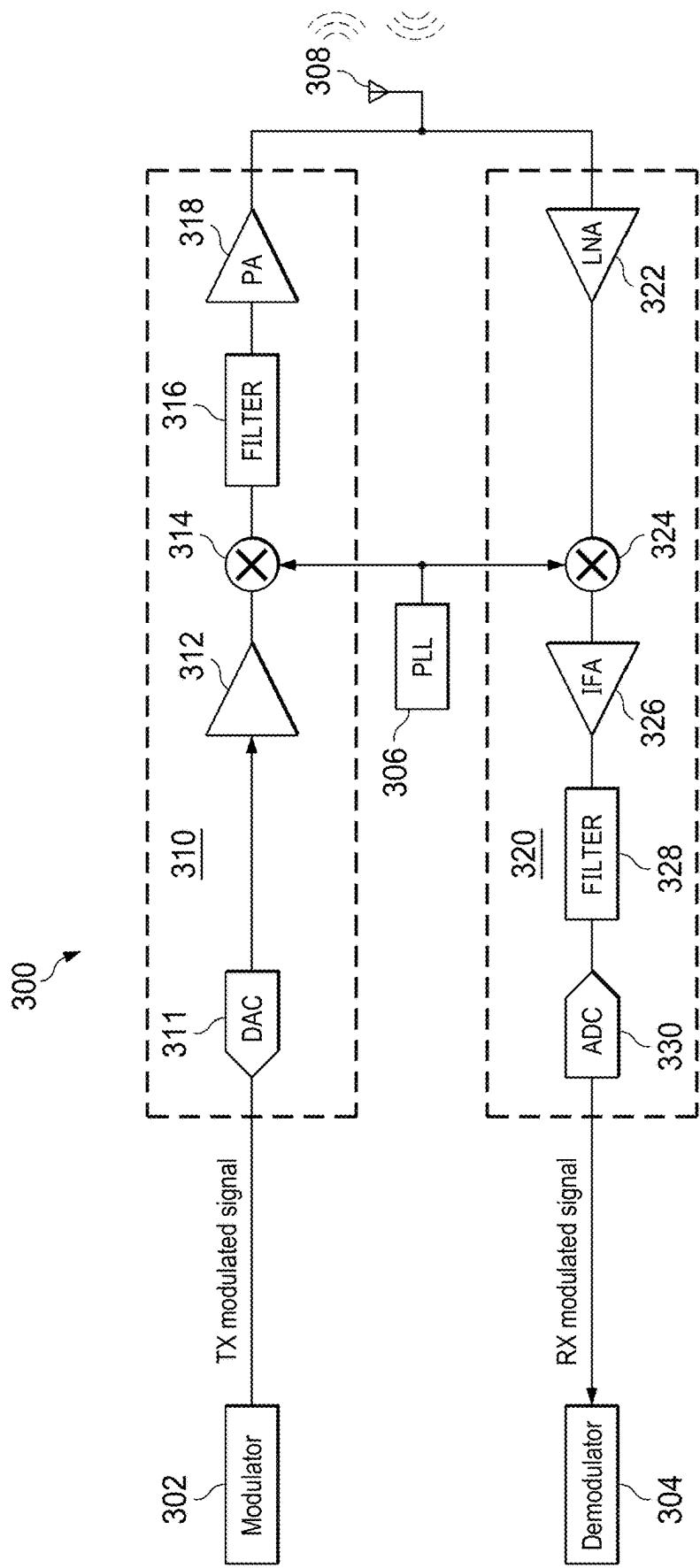


FIG. 3

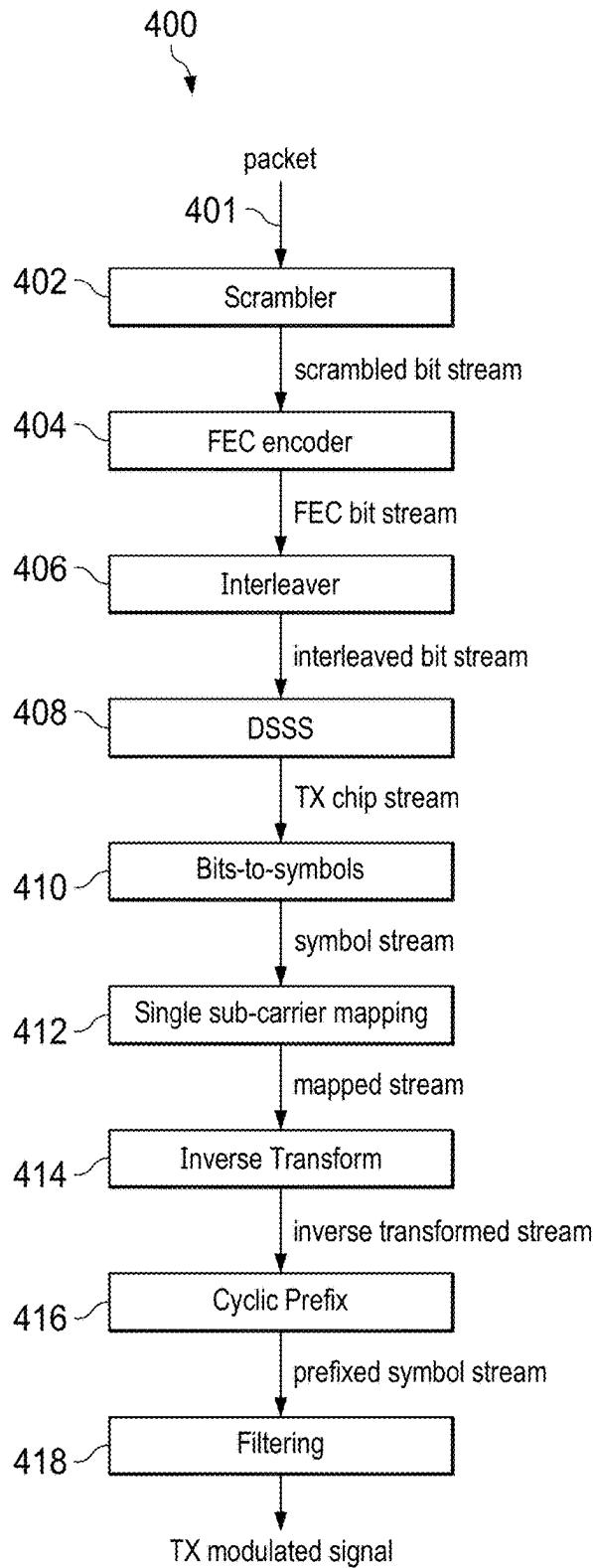


FIG. 4

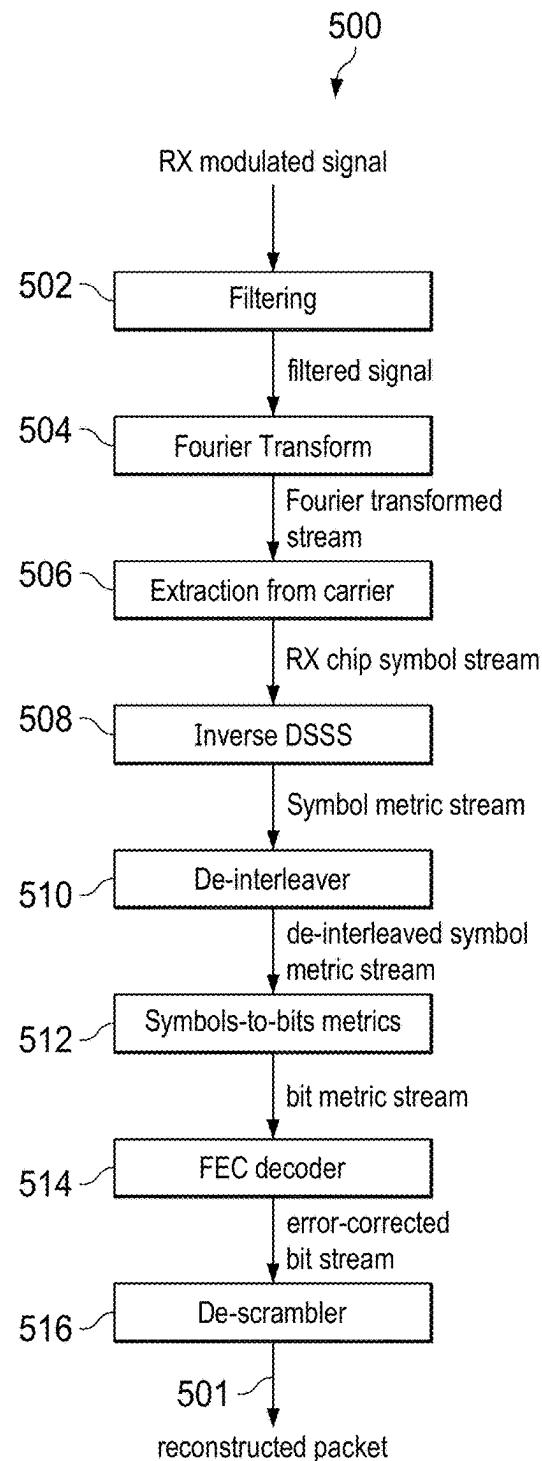


FIG. 5

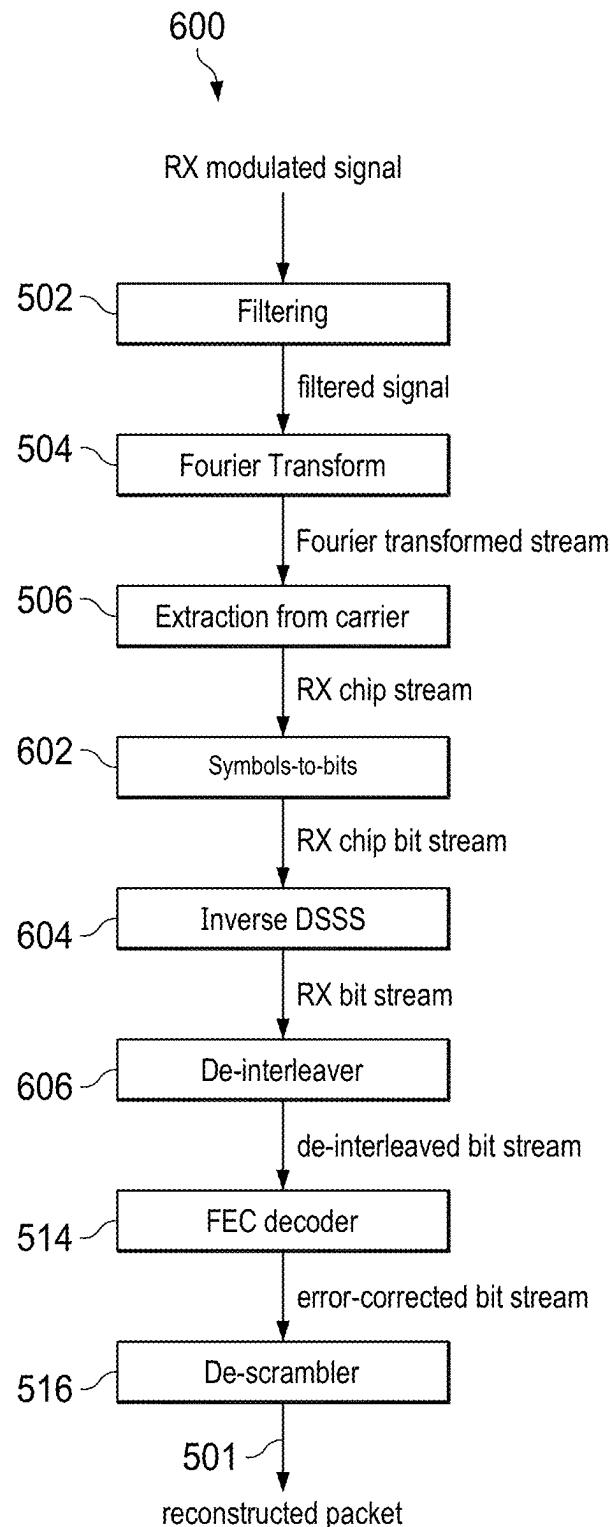


FIG. 6

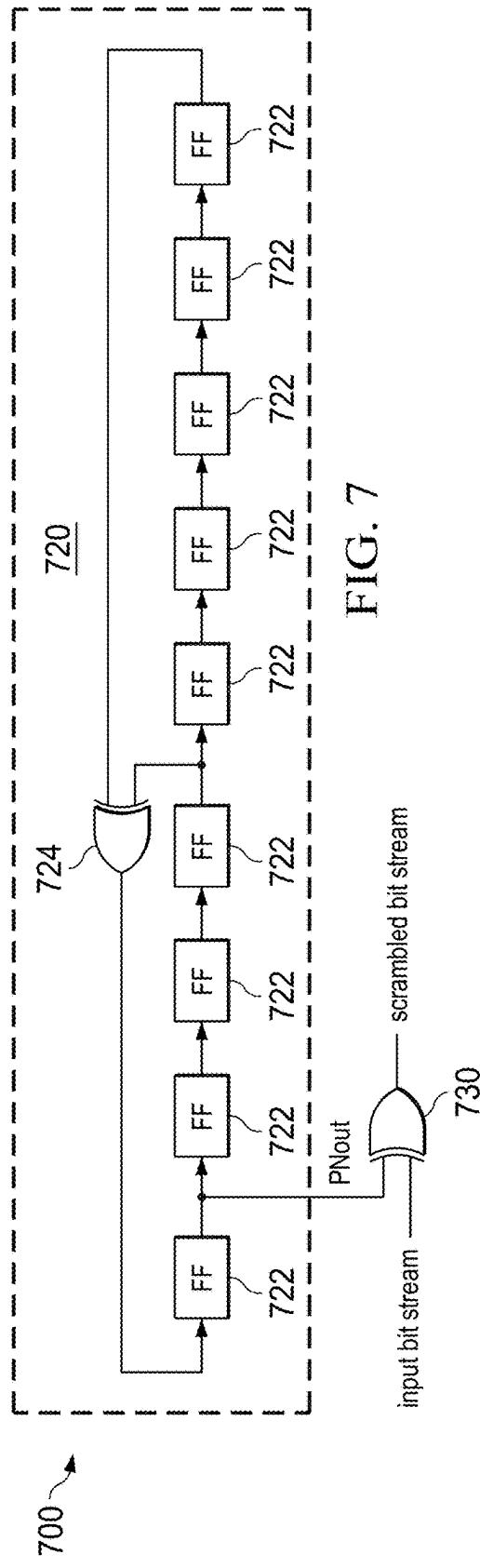


FIG. 7

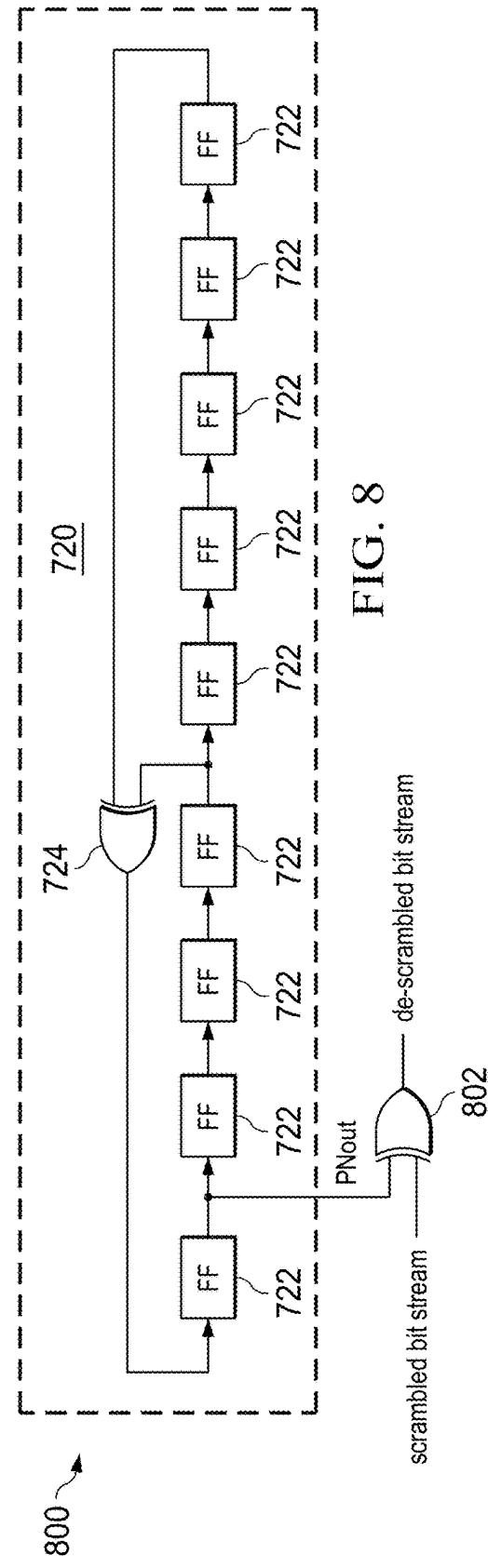


FIG. 8

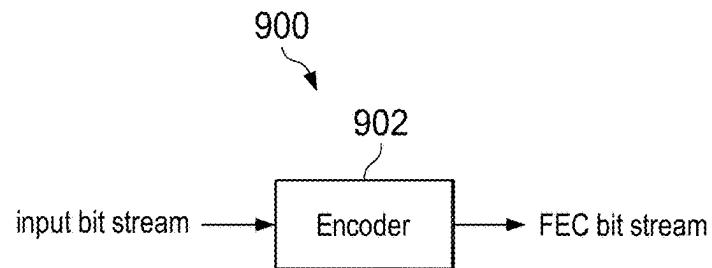
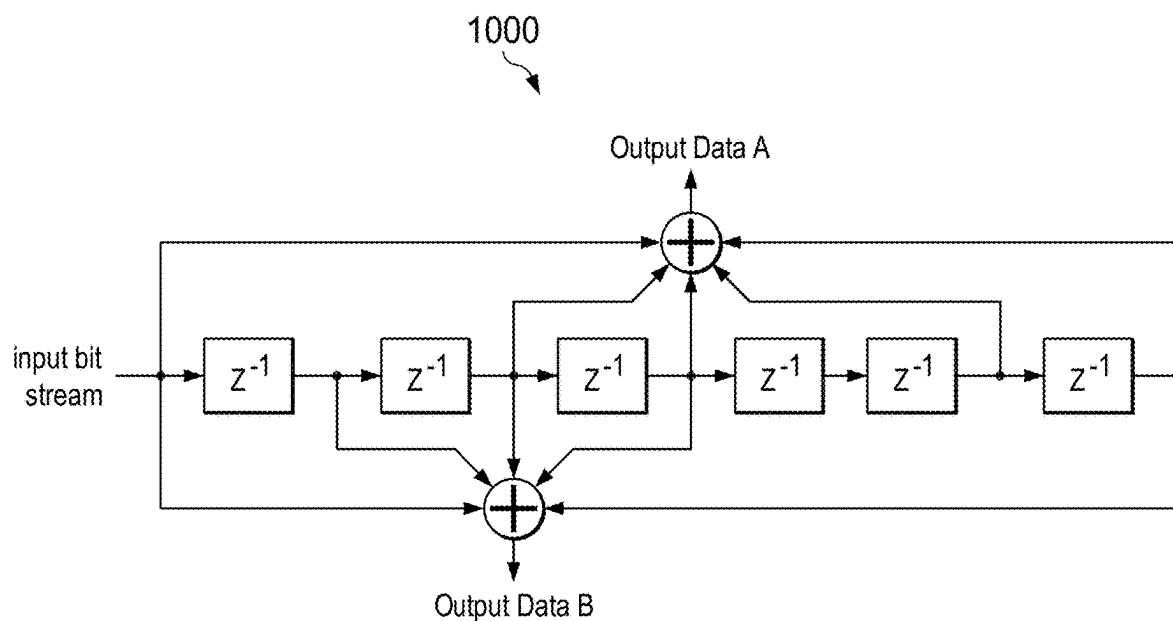


FIG. 9



Convolutional Encoder: Rate 1/2, constraint length K=7
Octal generator polynomials [133, 171]

FIG. 10

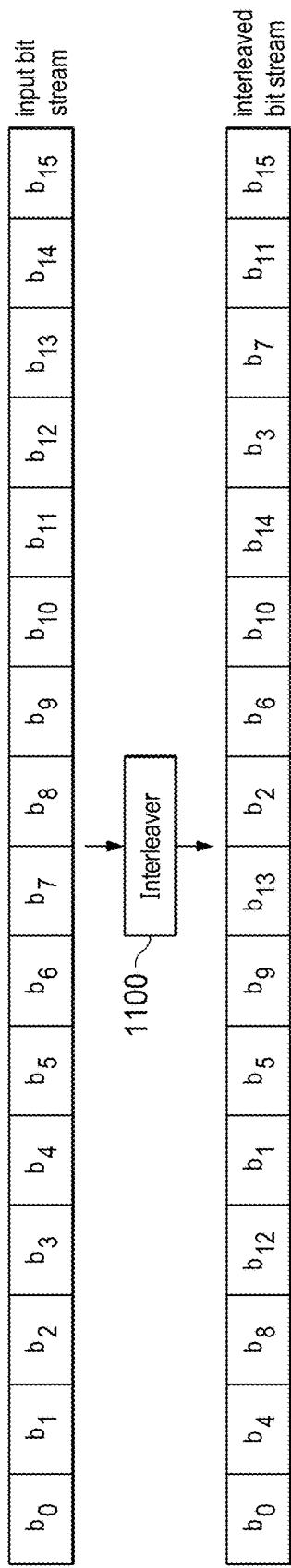


FIG. 11

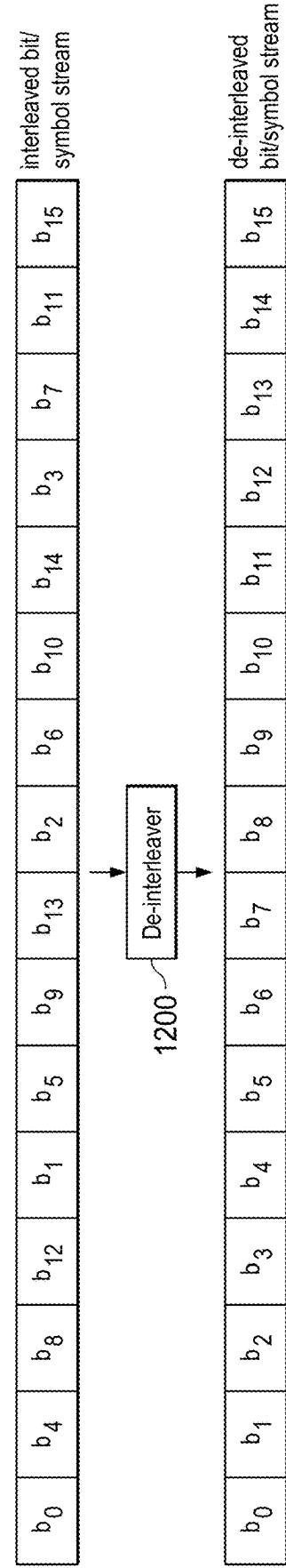


FIG. 12

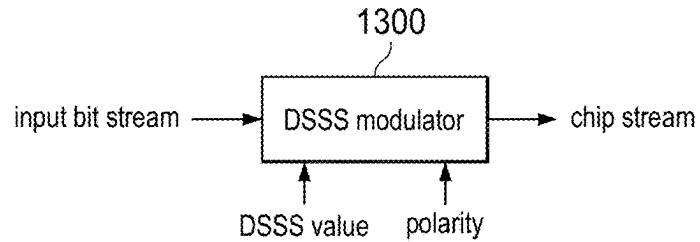


FIG. 13

DSSS value = 2	input bit	chip values [c0 c1]	DSSS value = 4	input bit	chip values [c0 c1 c2 c3]
polarity = even	0	0 0	polarity = even	0	0 0 1 1
polarity = even	1	0 1	polarity = even	1	0 1 1 0
polarity = odd	0	1 1	polarity = odd	0	1 1 0 0
polarity = odd	1	1 0	polarity = odd	1	1 0 0 1

DSSS value = 6	input bit	chip values [c0 c1 c2 c3]	DSSS value = 8	input bit	chip values [c0 .. c7]
polarity = even	0	0 0 1 1 0 0	polarity = even	0	0 0 1 1 0 0 1 1
polarity = even	1	0 1 1 0 1 0	polarity = even	1	0 1 1 0 1 0 0 1
polarity = odd	0	1 1 0 0 1 1	polarity = odd	0	1 1 0 0 1 1 0 0
polarity = odd	1	1 0 0 1 0 1	polarity = odd	1	1 0 0 1 0 1 1 0

DSSS value = 10	input bit	chip values [c0 .. c9]	DSSS value = 12	input bit	chip values [c0 .. c11]
polarity = even	0	0 0 1 1 0 0 1 1 1 1	polarity = even	0	0 0 1 1 0 0 1 1 1 1 0 0
polarity = even	1	0 1 1 0 1 0 0 1 1 0	polarity = even	1	0 1 1 0 1 0 0 1 1 0 0 1
polarity = odd	0	1 1 0 0 1 1 0 0 0 0	polarity = odd	0	1 1 0 0 1 1 0 0 0 0 1 1
polarity = odd	1	1 0 0 1 0 1 1 0 0 1	polarity = odd	1	1 0 0 1 0 1 1 0 0 1 1 0

FIG. 14

DSSS value = 2	input bit	chip values [c0 c1]	DSSS value = 4	input bit	chip values [c0 c1 c2 c3]
polarity = even	0	0 0	polarity = even	0	0 0 1 1
polarity = even	1	0 1	polarity = even	1	0 1 1 0
polarity = odd	0	1 1	polarity = odd	0	1 1 0 0
polarity = odd	1	1 0	polarity = odd	1	1 0 0 1

DSSS value = 6	input bit	chip values [c0 c1 c2 c3]	DSSS value = 8	input bit	chip values [c0 .. c7]
polarity = even	0	0 0 1 1 0 0	polarity = even	0	0 0 1 1 0 0 1 1
polarity = even	1	0 1 1 0 0 1	polarity = even	1	0 1 1 0 0 1 1 0
polarity = odd	0	1 1 0 0 1 1	polarity = odd	0	1 1 0 0 1 1 0 0
polarity = odd	1	1 0 0 1 1 0	polarity = odd	1	1 0 0 1 1 0 0 1

DSSS value = 10	input bit	chip values [c0 .. c9]	DSSS value = 12	input bit	chip values [c0 .. c11]
polarity = even	0	0 0 1 1 0 0 1 1 1 1	polarity = even	0	0 0 1 1 0 0 1 1 1 1 0 0
polarity = even	1	0 1 1 0 0 1 1 0 1 0	polarity = even	1	0 1 1 0 0 1 1 0 1 0 0 1
polarity = odd	0	1 1 0 0 1 1 0 0 0 0	polarity = odd	0	1 1 0 0 1 1 0 0 0 0 1 1
polarity = odd	1	1 0 0 1 1 0 0 1 0 1	polarity = odd	1	1 0 0 1 1 0 0 1 0 1 1 0

FIG. 15

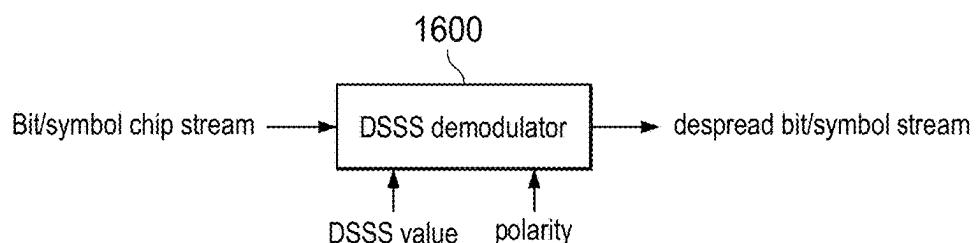


FIG. 16

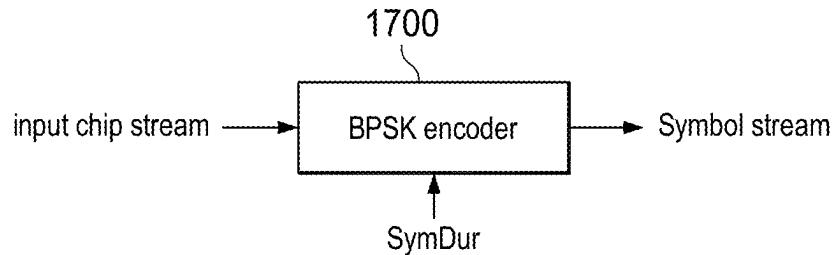


FIG. 17

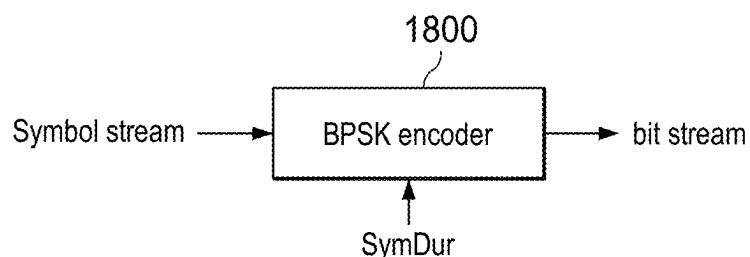


FIG. 18

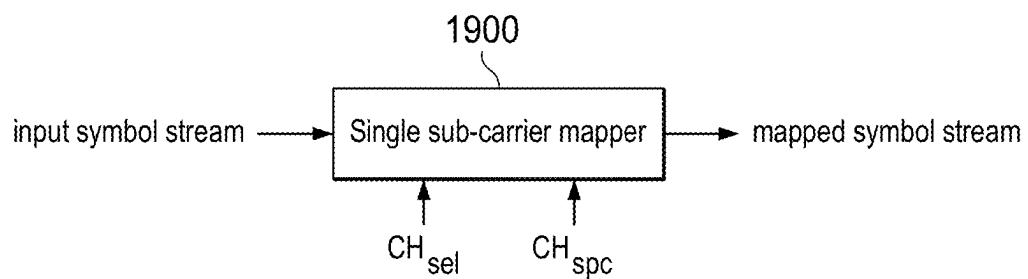


FIG. 19

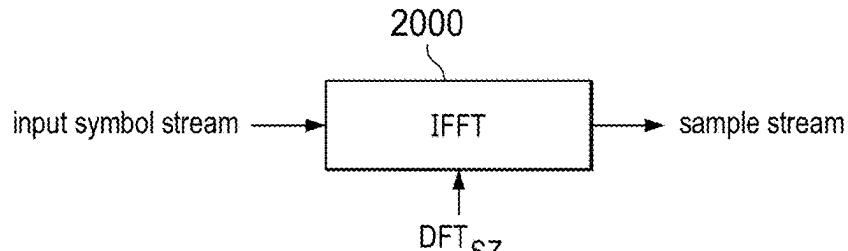


FIG. 20

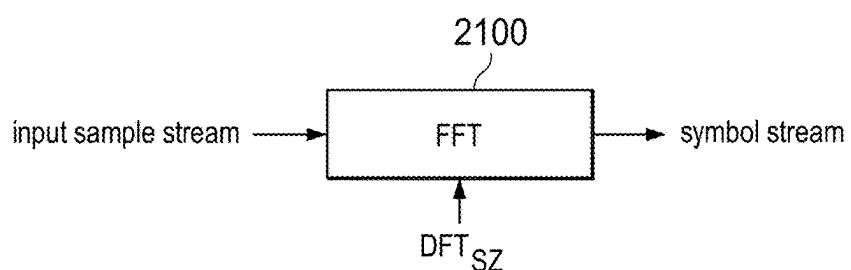


FIG. 21

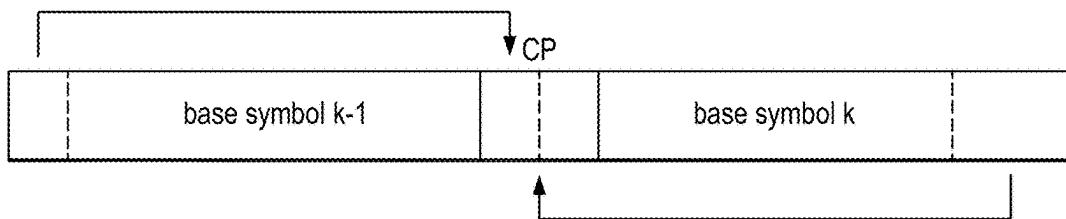


FIG. 22

SymDur = 120 μ s					
Parameter	Option 1	Option 2	Option 3	Option 4	
Nominal BW [kHz]	1100	550	280	156	
Channel spacing (CH_{spc}) [kHz]	1200	800	400	200	
DFT size (DFT_{SZ}) [samples]	128	64	32	16	
Active tones	104	52	26	12	
Data rate [kb/s]	DSSS Value = 2	2.083	2.083	2.083	2.083
	DSSS Value = 4	1.041	1.041	1.041	1.041
	DSSS Value = 6	0.694	0.694	0.694	0.694

FIG. 23A

SymDur = 60 μ s					
Parameter	Option 1	Option 2	Option 3		
Nominal BW [kHz]	1100	550	280		
Channel spacing (CH_{spc}) [kHz]	1200	800	400		
DFT size (DFT_{SZ}) [samples]	64	32	16		
Active tones	52	26	12		
Data rate [kb/s]	DSSS Value = 2	4.167	4.167	4.167	
	DSSS Value = 4	2.083	2.083	2.083	
	DSSS Value = 6	1.389	1.389	1.389	

FIG. 23B

SymDur = 30 μ s			
Parameter	Option 1	Option 2	
Nominal BW [kHz]	1100	550	
Channel spacing (CH_{spc}) [kHz]	1200	800	
DFT size (DFT_{SZ}) [samples]	32	16	
Active tones	26	12	
Data rate [kb/s]	DSSS Value = 2	8.333	8.333
	DSSS Value = 4	4.167	4.167
	DSSS Value = 6	2.778	2.778

FIG. 23C

SymDur = 15 μ s		
Parameter	Option 1	
Nominal BW [kHz]	1100	
Channel spacing (CH_{spc}) [kHz]	1200	
DFT size (DFT_{SZ}) [samples]	16	
Active tones	12	
Data rate [kb/s]	DSSS Value = 2	16.667
	DSSS Value = 4	8.333
	DSSS Value = 6	5.556

FIG. 23D

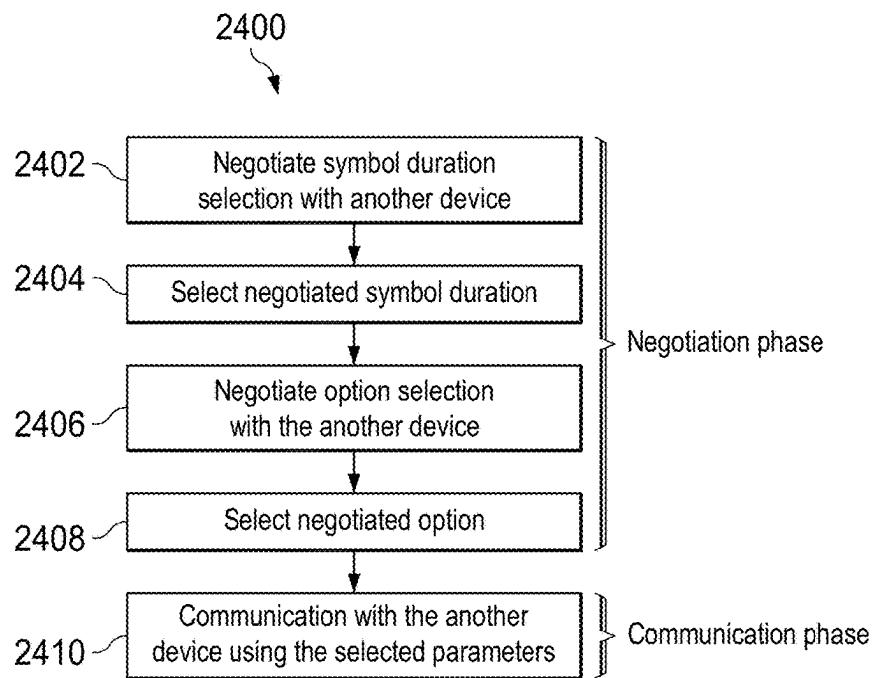


FIG. 24

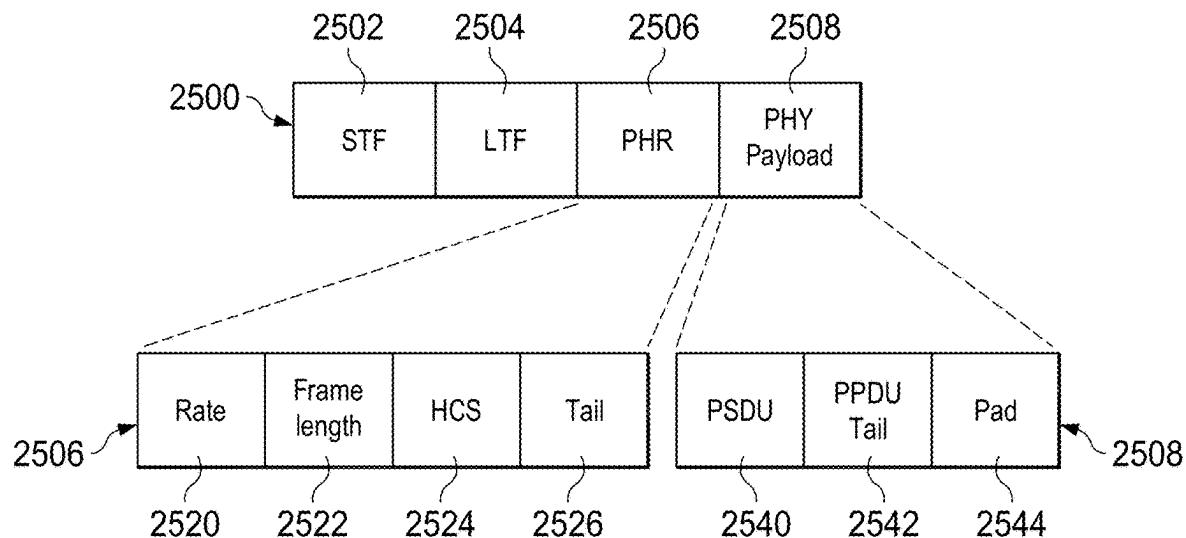


FIG. 25

2600



Index	Value	Index	Value	Index	Value	Index	Value
0	0	20	0	40	0	60	0
1	0	21	1	41	0	61	0
2	0	22	1	42	1	62	0
3	0	23	0	43	1	63	1
4	0	24	1	44	0	64	1
5	0	25	1	45	0	65	1
6	0	26	1	46	0	66	0
7	0	27	0	47	0	67	0
8	0	28	0	48	0	68	1
9	0	29	1	49	0	69	0
10	0	30	0	50	1	70	1
11	0	31	0	51	1	71	0
12	0	32	0	52	1	72	0
13	0	33	1	53	1	73	0
14	0	34	0	54	1	74	1
15	0	35	1	55	0	75	0
16	1	36	1	56	0	76	1
17	0	37	0	57	1	77	0
18	0	38	1	58	1	78	1
19	1	39	1	59	1	79	0

FIG. 26

2700
↓

Index	Value	Index	Value	Index	Value
0	0	12	0	24	0
1	0	13	0	25	1
2	0	14	0		
3	0	15	0		
4	1	16	1		
5	1	17	0		
6	1	18	1		
7	1	19	1		
8	0	20	0		
9	1	21	0		
10	1	22	1		
11	1	23	1		

FIG. 27

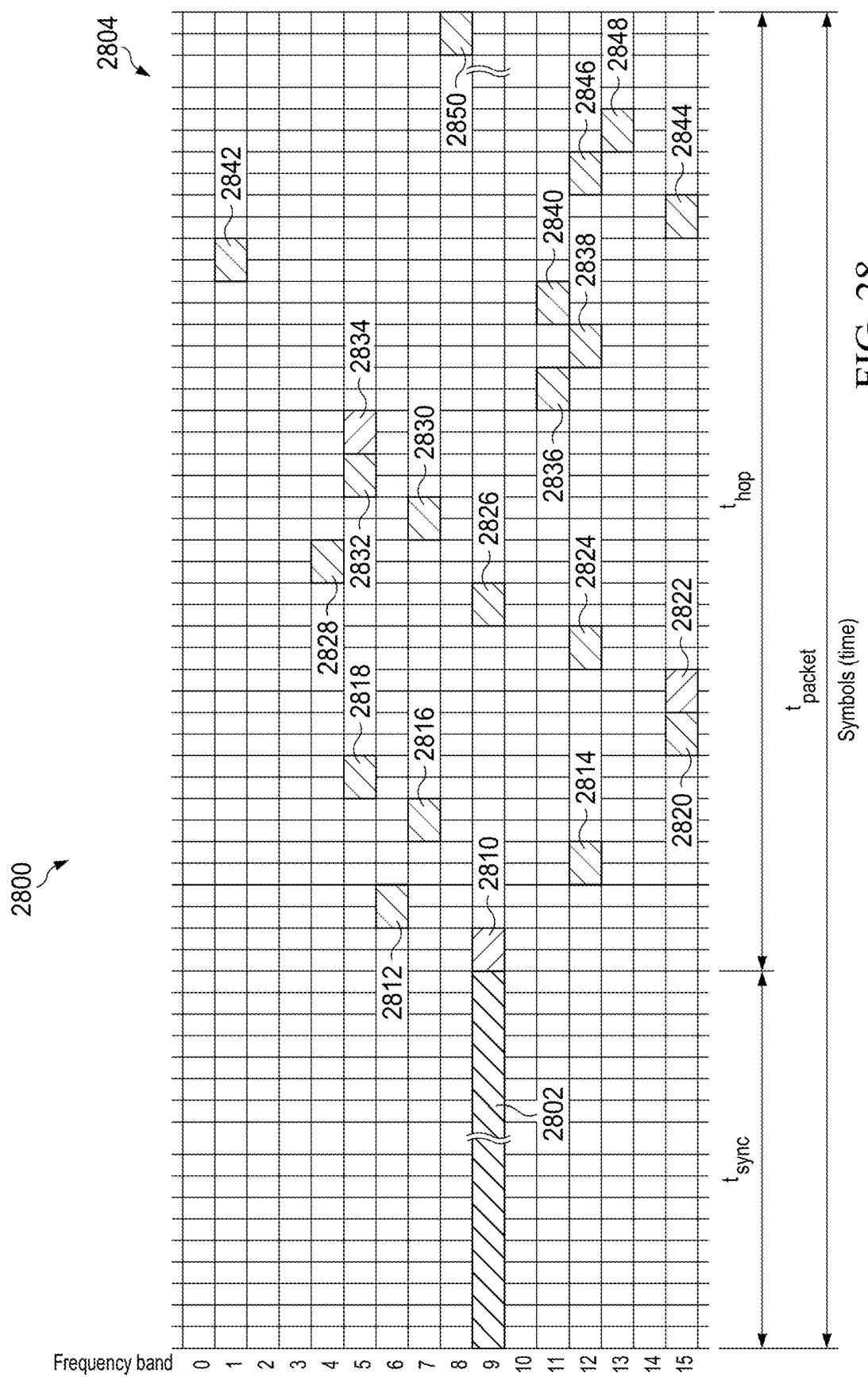


FIG. 28

Index	LCGa
0	17
1	29
2	37
3	41
4	53
5	61
6	73
7	89

FIG. 29A

Index	Value	Index	Value	Index	Value
0	3	10	37	20	79
1	5	11	41	21	83
2	7	12	43	22	89
3	11	13	47	23	97
4	13	14	53	24	101
5	17	15	59	25	103
6	19	16	61	26	107
7	23	17	67	27	109
8	29	18	71	28	113
9	31	19	73	29	127

FIG. 29B

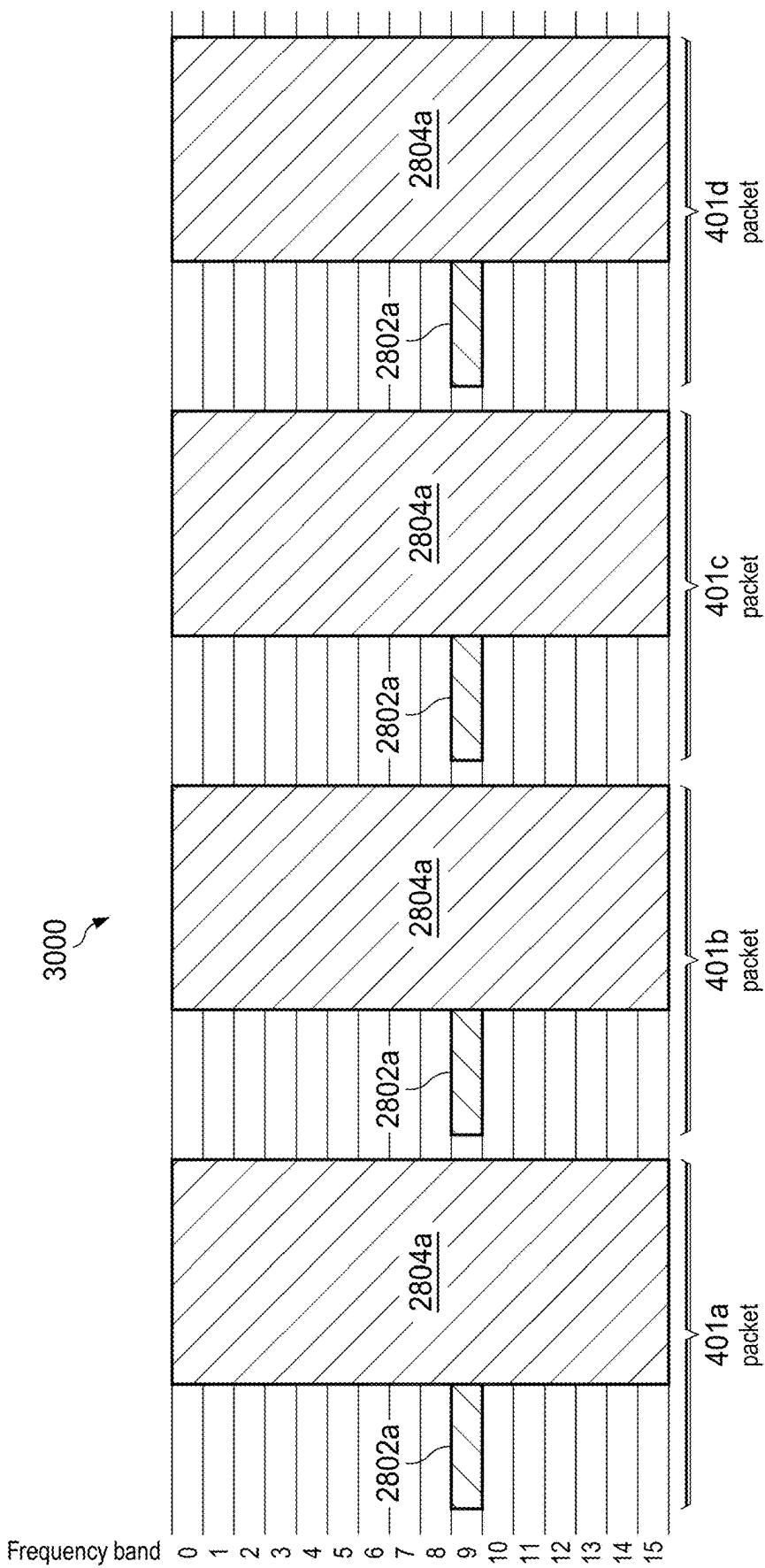


FIG. 30

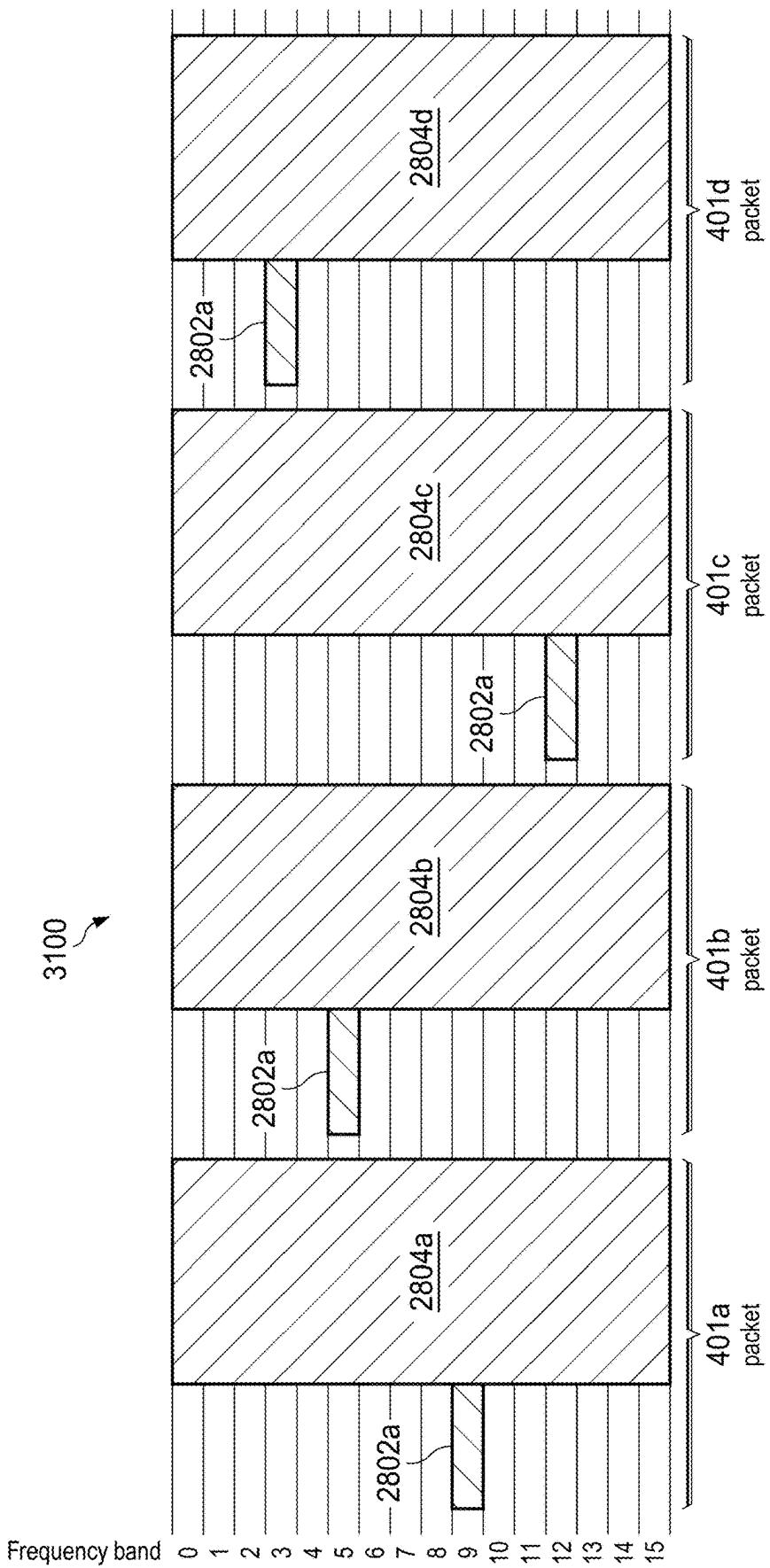


FIG. 31

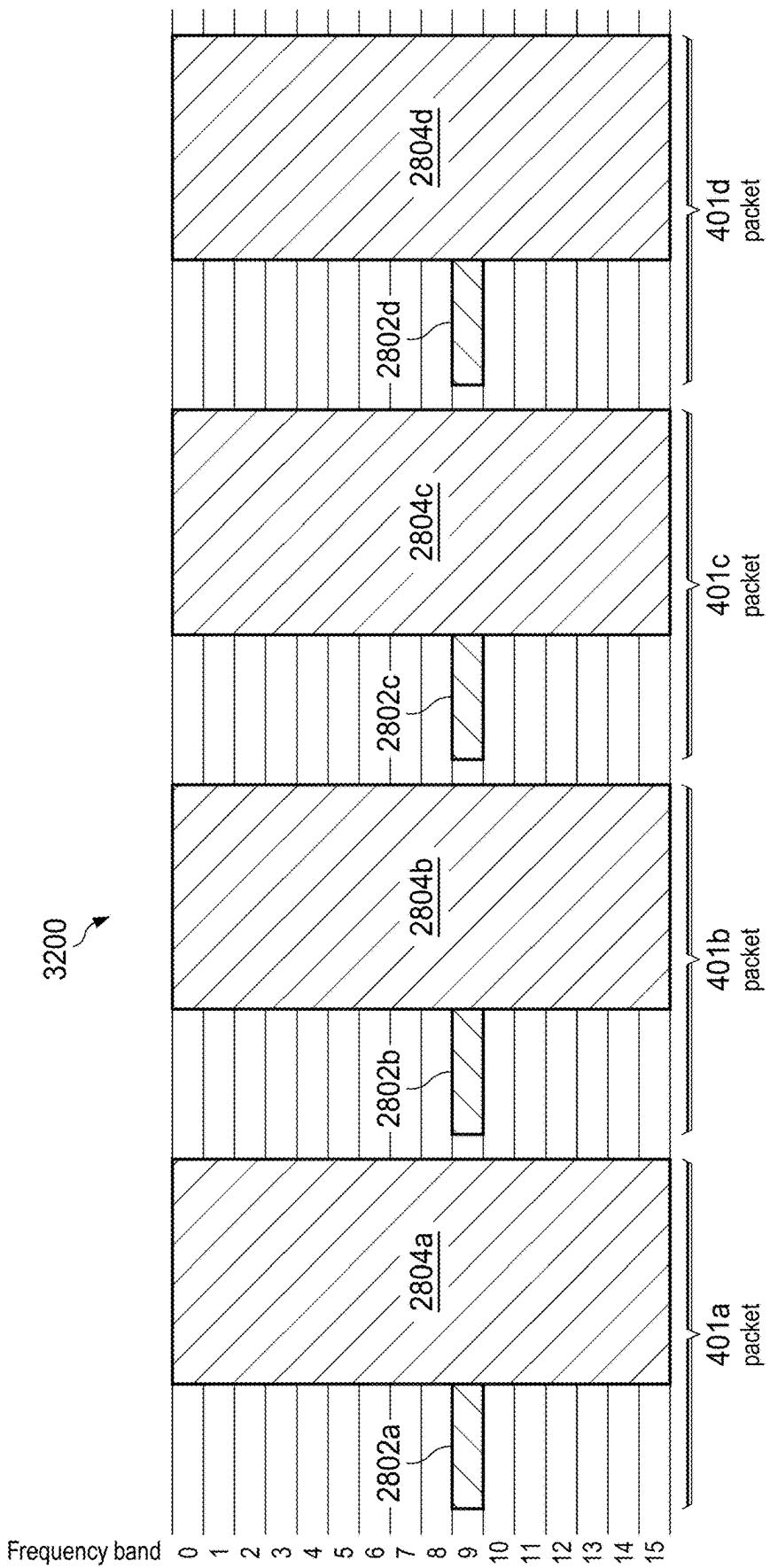


FIG. 32

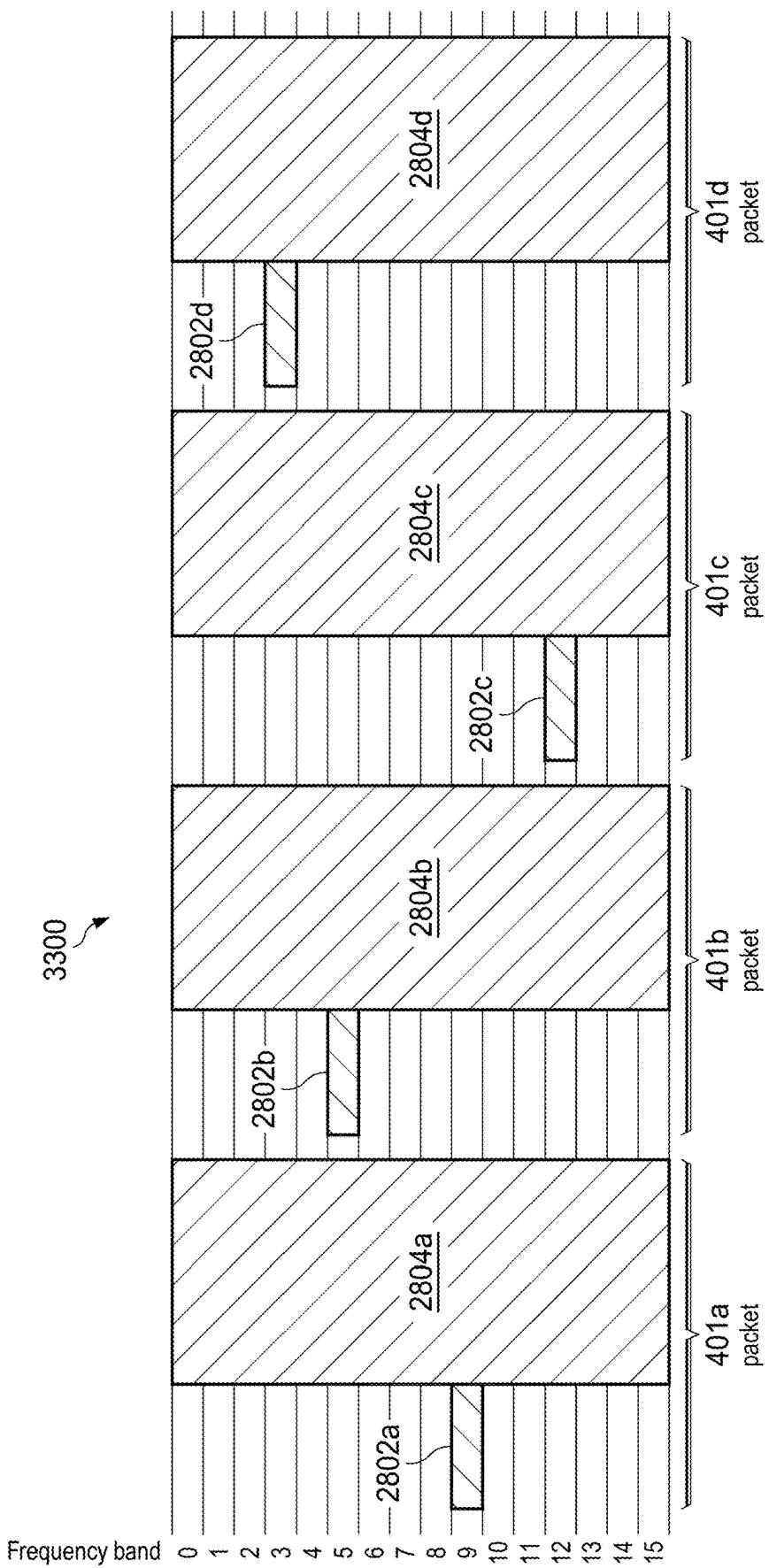


FIG. 33

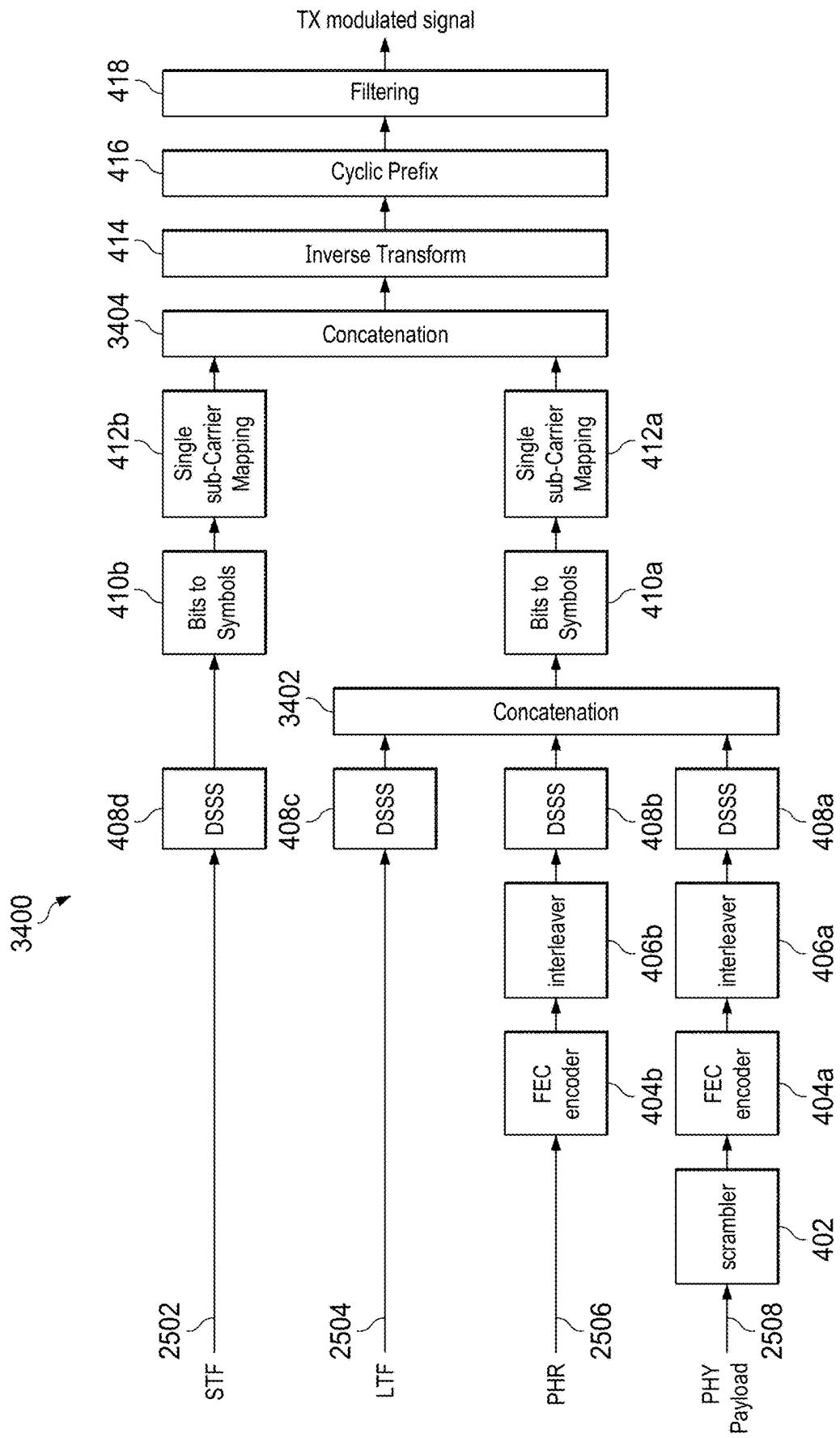


FIG. 34

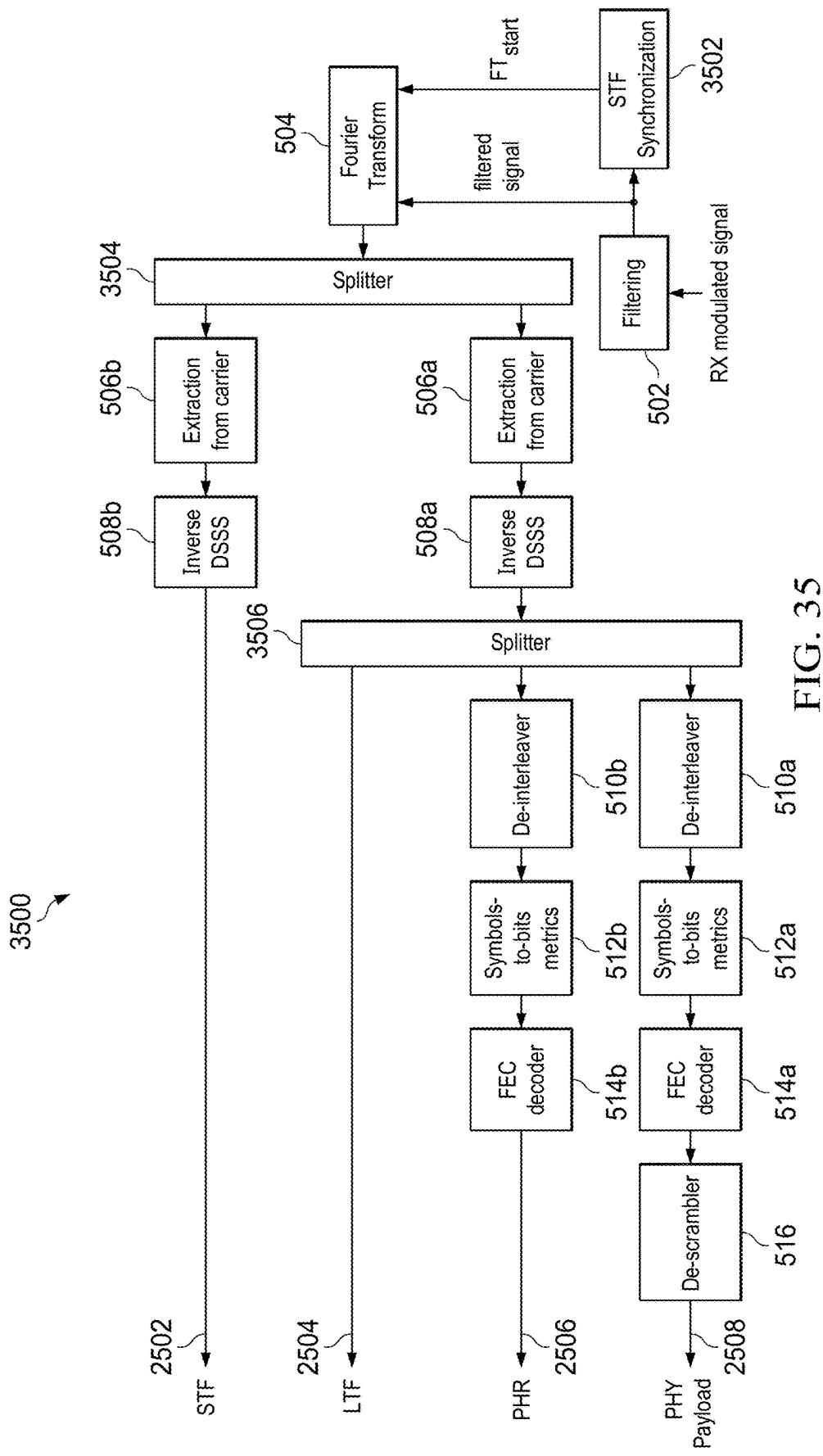


FIG. 35

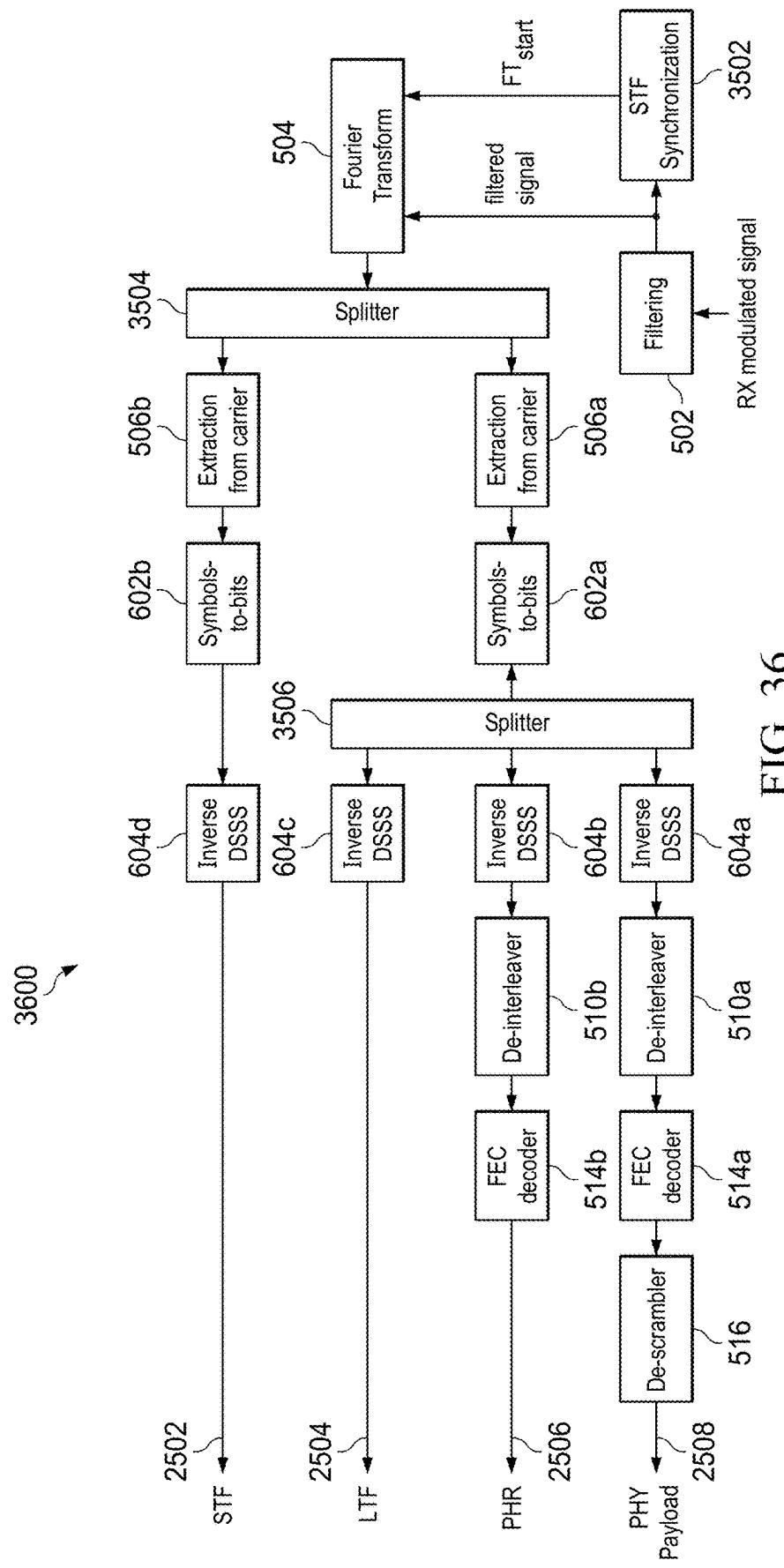


FIG. 36

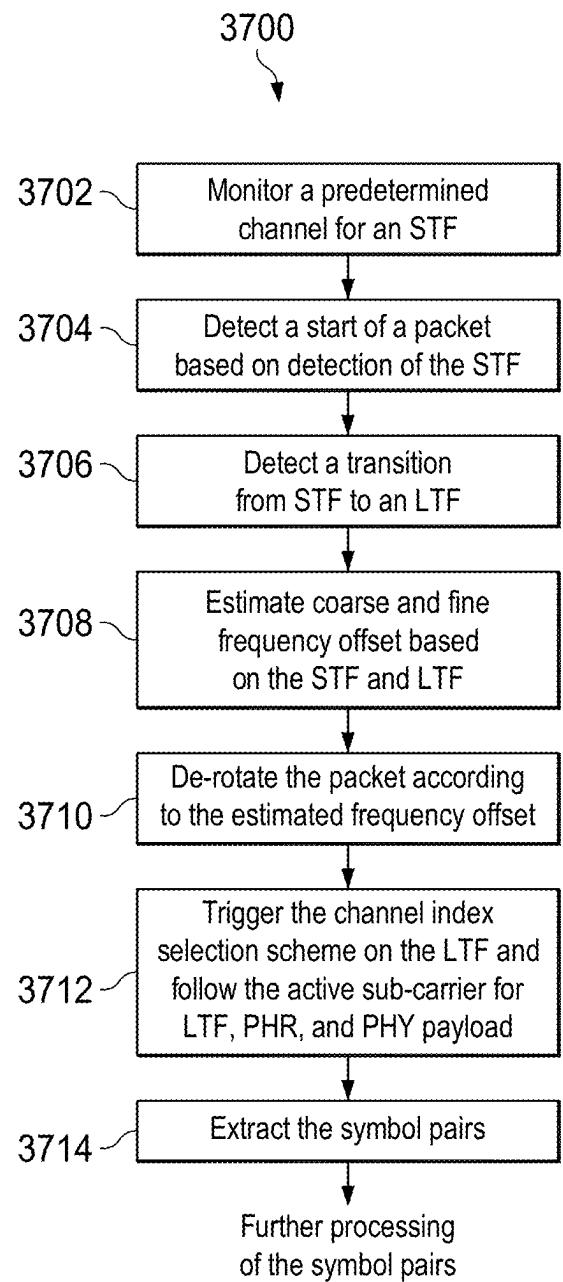


FIG. 37

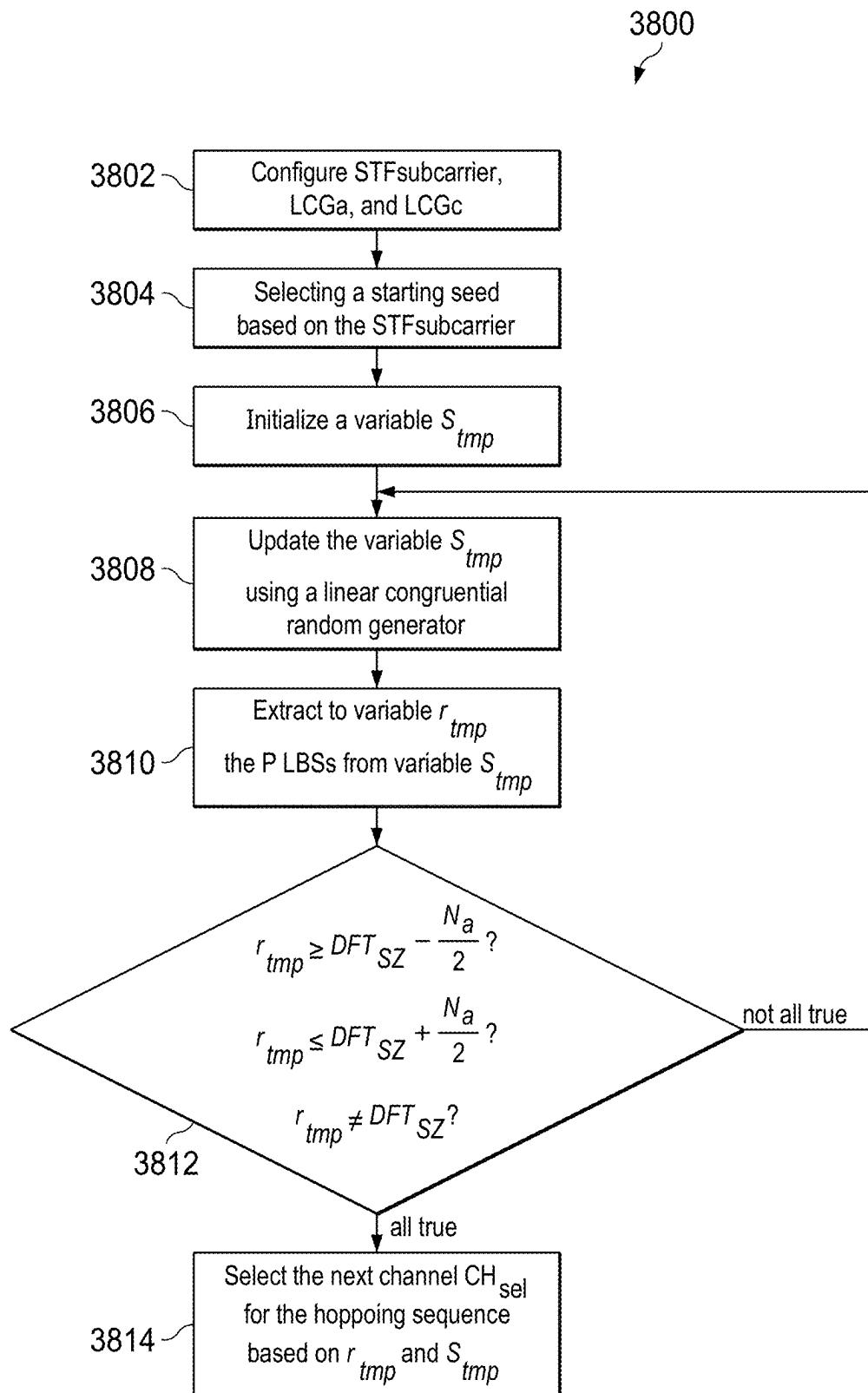


FIG. 38

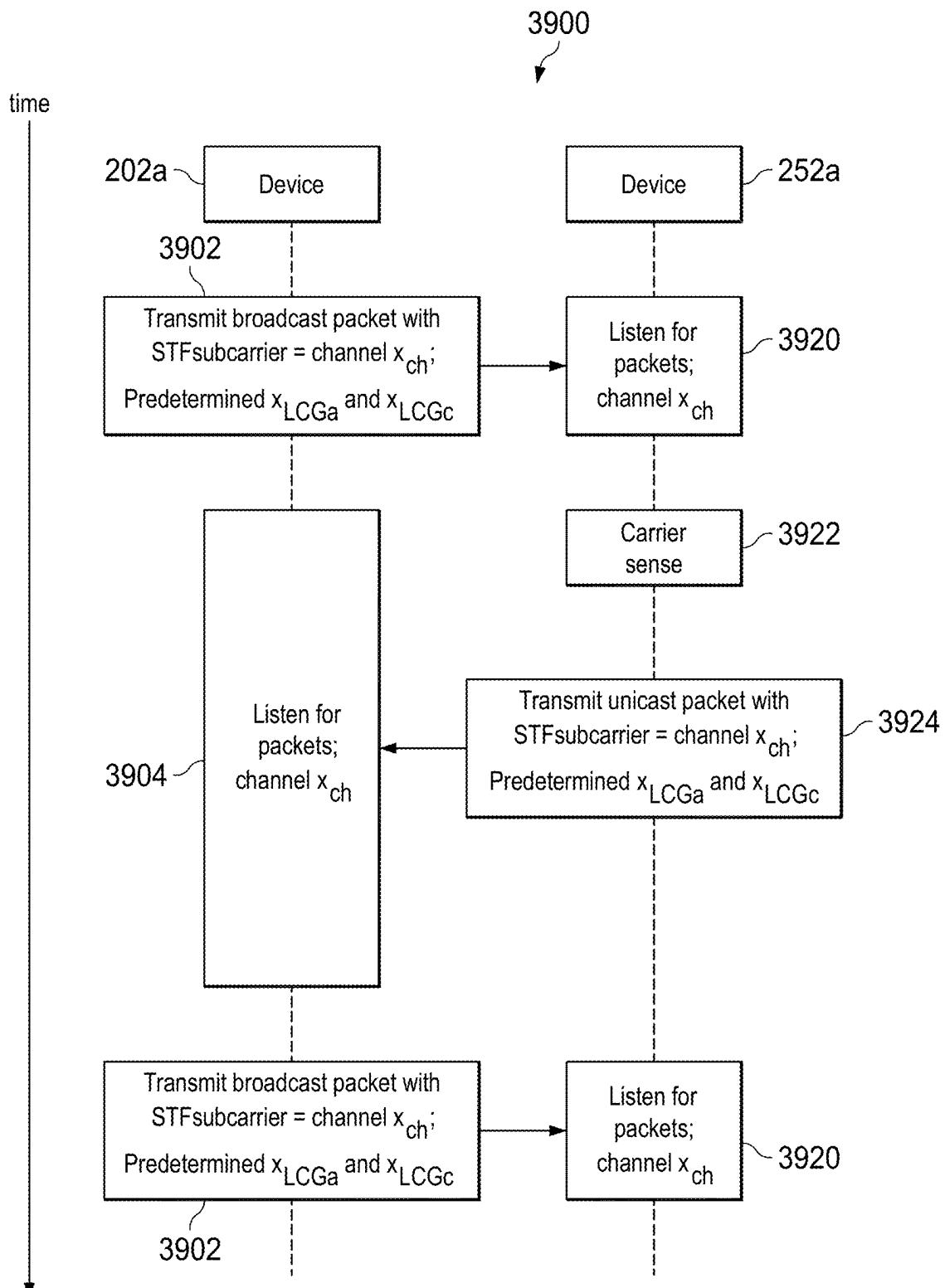


FIG. 39

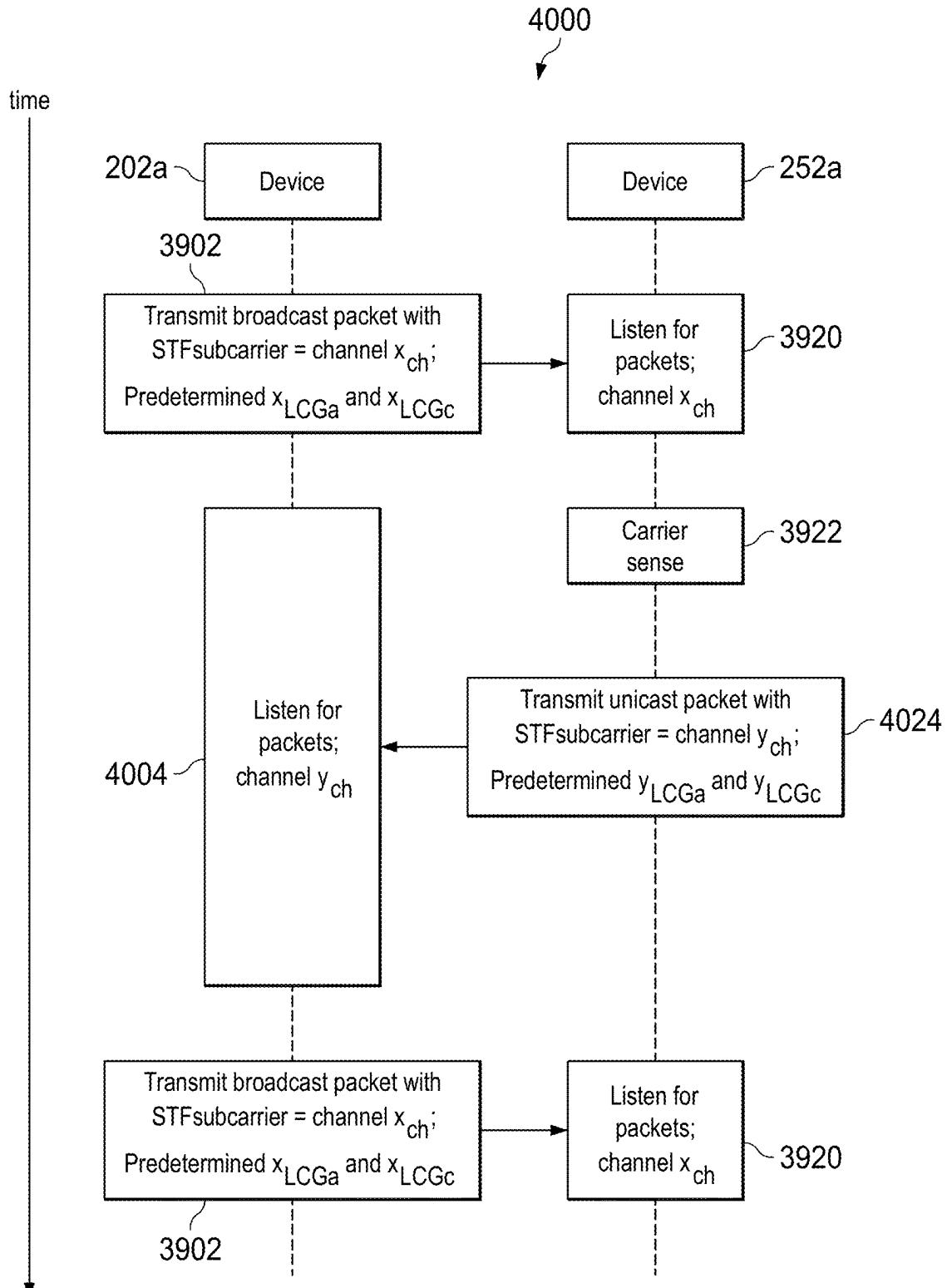


FIG. 40

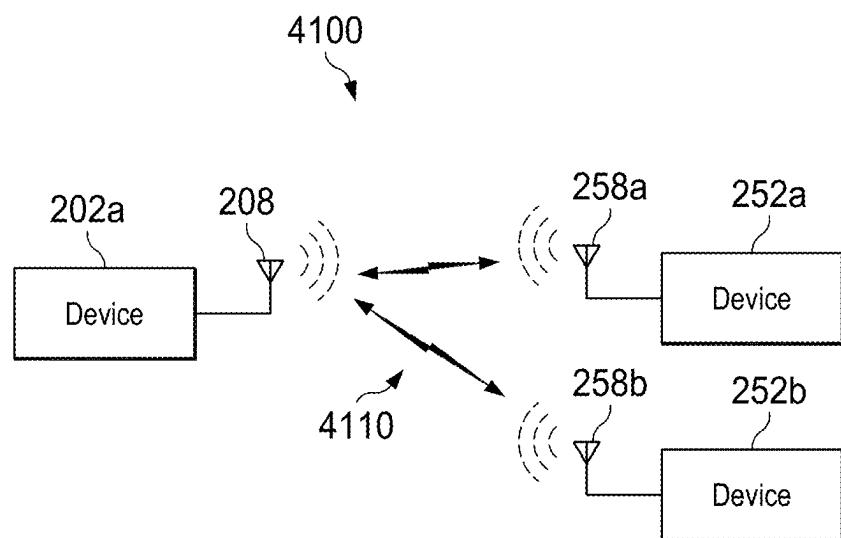


FIG. 41

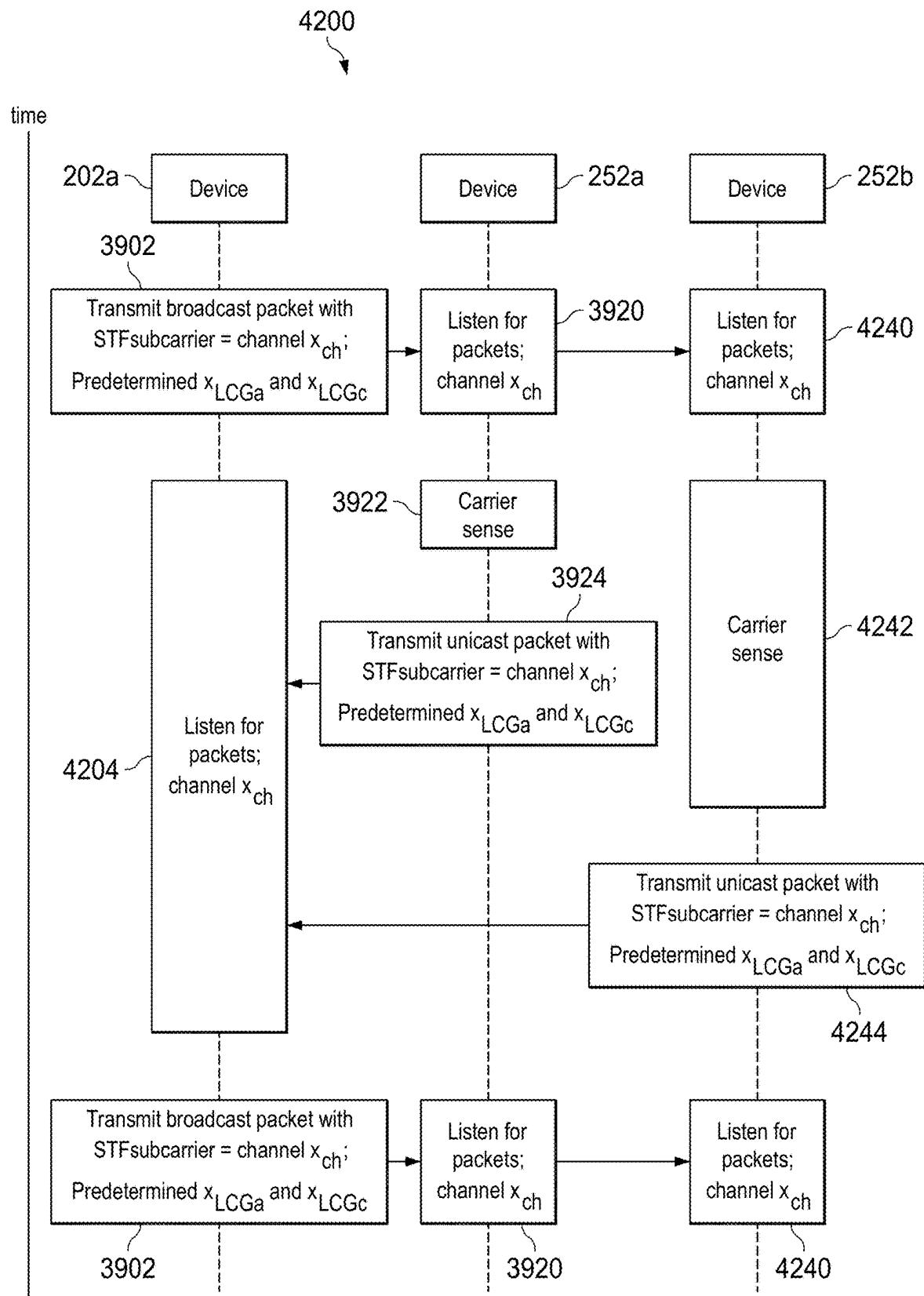


FIG. 42

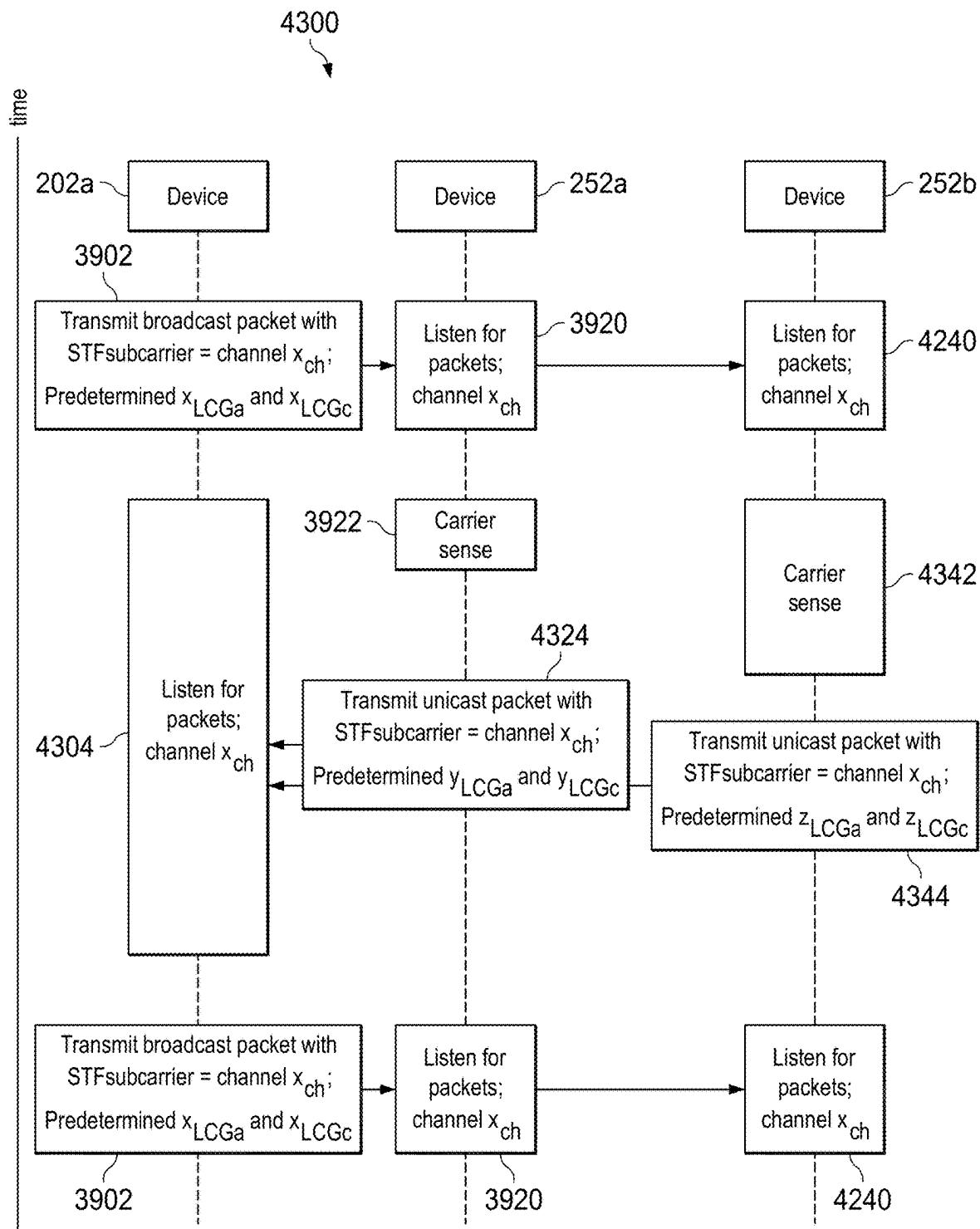


FIG. 43

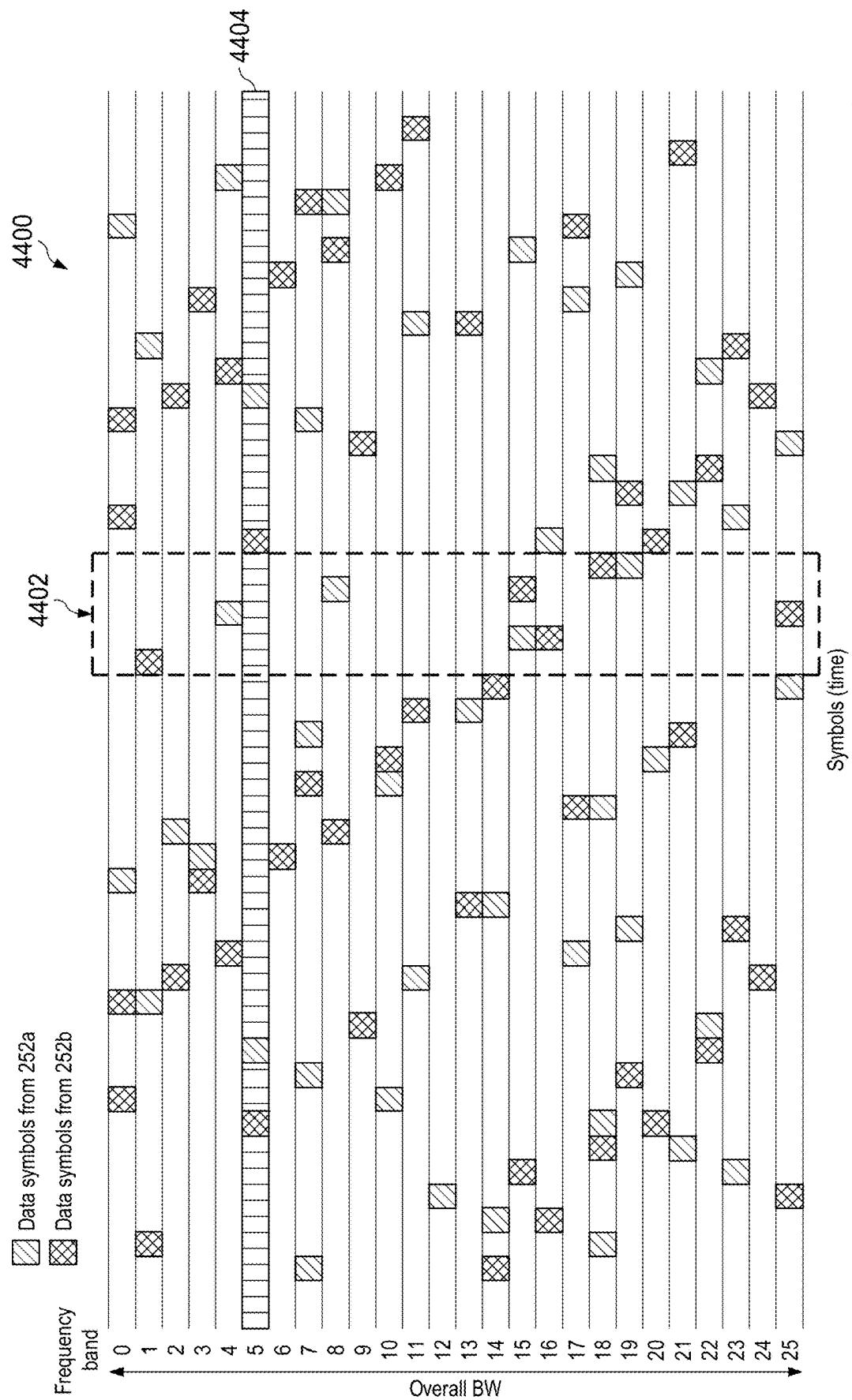


FIG. 44

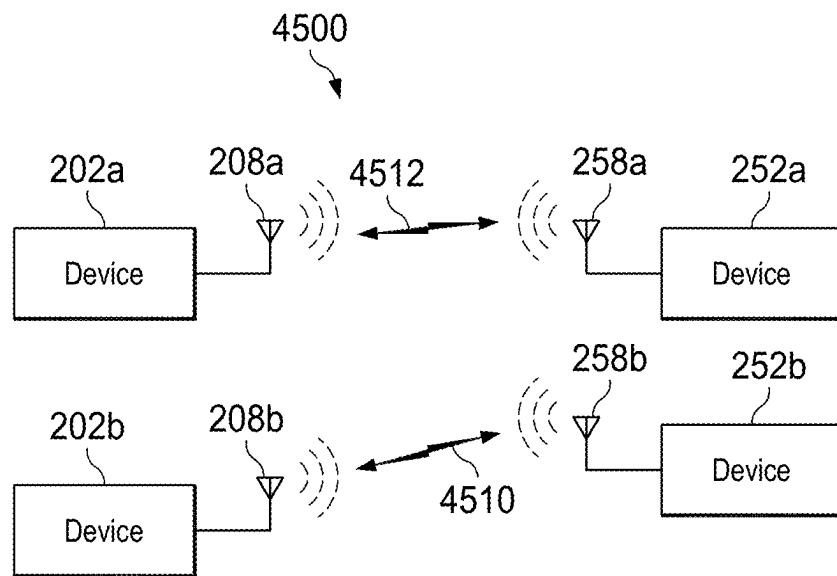


FIG. 45

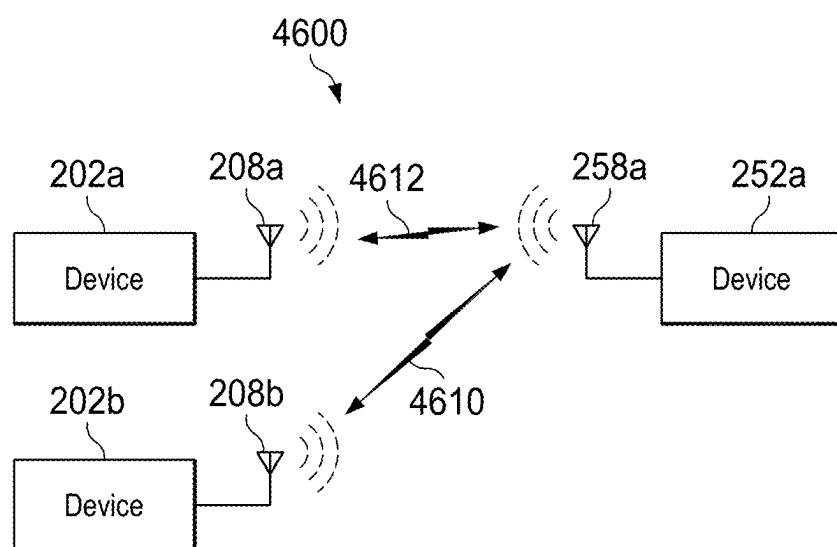


FIG. 46

OFDM COMMUNICATION WITH HOPPING SEQUENCE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 63/555,011, filed Feb. 17, 2024, which application is hereby incorporated herein by reference.

TECHNICAL FIELD

[0002] The present disclosure relates generally to an electronic system and method, and, in particular embodiments, to orthogonal frequency-division multiplexing (OFDM) communication with hopping sequence.

BACKGROUND

[0003] In frequency-division multiplexing (FDM), a transmitter can encode data in multiple frequency bands and transmit a radio-frequency (RF) signal that combines the signals from those frequency bands. The RF signal is a combination of multiple sub-carrier signals, each of which encodes information. Because unique information can be encoded in each frequency band, an FDM system typically has a higher data throughput relative to other systems that use only one carrier frequency.

[0004] In an orthogonal FDM (OFDM) system, each frequency band is orthogonal to the adjacent frequency bands (e.g., a center frequency of a first frequency band aligns with the null frequency of each adjacent frequency band). The orthogonality of the frequency bands may result in reduced interference across the carrier signals. As such, in an OFDM system, spacing between the center frequency of each sub-channel (also referred to as channel, sub-carrier band, or frequency band) can be closer to each other (e.g., sub-channels may overlap in frequency) than in an FDM system without orthogonality, where sub-channels do not overlap in frequency, as illustrated in FIG. 1. In addition, the orthogonality may allow for an OFDM receiver to more easily extract information from each frequency band of the combined RF signal.

SUMMARY

[0005] In accordance to an embodiment, a method includes: transmitting, by a first device, a first synchronization sequence in a single synchronization channel of a plurality of channels; and after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0006] In accordance to an embodiment, a method includes: monitoring, by a first device, a single synchronization channel of a plurality of channels for detection of a first synchronization sequence; and responsive to detecting the first synchronization sequence, extracting, by the first device, first data symbols associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0007] In accordance to an embodiment, a method includes: generating, by a first device, a first plurality of spreads of chips corresponding to a plurality of bits of a synchronization sequence of a packet; generating, by the

first device, a second plurality of spreads of chips corresponding to a plurality of bits of a header field of the packet; generating, by the first device, a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; generating, by the first device, a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; transmitting, by the first device, the first plurality of pairs of BPSK symbols in a single synchronization channel of a plurality of channels; and after transmitting the first plurality of pairs of BPSK symbols, transmitting, by the first device, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission at a time.

[0008] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single synchronization channel of a plurality of channels, and after transmitting the first synchronization sequence, transmit first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0009] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: monitor, using the transceiver, a single synchronization channel of a plurality of channels for detection of a first synchronization sequence, and responsive to detecting the first synchronization sequence, extract first data symbols associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0010] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: generate a first plurality of spreads of chips corresponding to a plurality of bits of a synchronization sequence of a packet, generate a second plurality of spreads of chips corresponding to a plurality of bits of a header field of the packet, generate a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips, generate a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips, transmit, via the transceiver, the first plurality of pairs of BPSK symbols in a single synchronization channel of a plurality of channels, and after transmitting the first plurality of pairs of BPSK symbols, transmit, via the transceiver, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission at a time.

[0011] In accordance to an embodiment, a method includes: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; after transmitting the first data, transmitting, by the first device, a second synchronization sequence in a single second synchronization channel of the plurality of channels; and after transmitting the second synchronization sequence, transmitting, by the first device, second data associated with the second synchronization sequence according to a second hopping sequence using a

single channel of the plurality of channels at a time, where: the first synchronization channel is different from the second synchronization channel, or the first synchronization sequence is different from the second synchronization sequence, or the second hopping sequence is different from the first hopping sequence.

[0012] In accordance to an embodiment, a method includes: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channel; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data, monitoring, by the first device, a single second synchronization channel for detection of a second synchronization sequence, where the first synchronization channel is different from the second synchronization channel.

[0013] In accordance to an embodiment, a method includes: generating a plurality of spreads of chips corresponding to a plurality of bits of a first packet; generating a plurality of binary phase shift keying (BPSK) symbols based on the plurality of spreads of chips; repeating a portion of each respective BPSK symbol of the plurality of BPSK symbols before the respective BPSK symbol; applying a windowing filter during a first portion of the repeated portion of each BPSK symbol to generate a second plurality of BPSK symbols; and transmitting the second plurality of BPSK symbols, as pairs of symbols, in respective channels of a plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission of the second plurality of BPSK symbols at a time.

[0014] In accordance to an embodiment, a method includes: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channel; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data: monitoring, by the first device, a single second synchronization channel for detection of a second synchronization sequence associated with a second device, and monitoring, by the first device, the single second synchronization channel for detection of a third synchronization sequence associated with a third device.

[0015] In accordance to an embodiment, a method includes: generating a first plurality of spreads of chips corresponding to a plurality of bits of a first packet; generating a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; transmitting the first plurality of pairs of BPSK symbols in respective channels of a plurality of channels according to a first hopping sequence; generating a second plurality of spreads of chips corresponding to a plurality of bits of a second packet; generating a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; and transmitting the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a second hopping sequence, where the second hopping sequence is different from the first hopping sequence, and where only a single channel of the plurality of channels is used for transmission at a time.

[0016] In accordance to an embodiment, a method includes: negotiating, by a first device with a second device, a first symbol duration; transmitting, by the first device, a first synchronization sequence in a single synchronization channel of a plurality of channels using the negotiated first symbol duration; and after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time using the negotiated symbol duration.

[0017] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a plurality of channel; after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; after transmitting the first data, transmit, via the transceiver, a second synchronization sequence in a single second synchronization channel of the plurality of channels; and after transmitting the second synchronization sequence, transmit, via the transceiver, second data associated with the second synchronization sequence according to a second hopping sequence using a single channel of the plurality of channels at a time, where: the first synchronization channel is different from the second synchronization channel, or the first synchronization sequence is different from the second synchronization sequence, or the second hopping sequence is different from the first hopping sequence.

[0018] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a plurality of channel; after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data, monitor a single second synchronization channel for detection of a second synchronization sequence, where the first synchronization channel is different from the second synchronization channel.

[0019] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: generate a plurality of spreads of chips corresponding to a plurality of bits of a first packet; generate a plurality of binary phase shift keying (BPSK) symbols based on the plurality of spreads of chips; repeat a portion of each respective BPSK symbol of the plurality of BPSK symbols before the respective BPSK symbol; apply a windowing filter during a first portion of the repeated portion of each BPSK symbol to generate a second plurality of BPSK symbols; and transmit, via the transceiver, the second plurality of BPSK symbols, as pairs of symbols, in respective channels of a plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission of the second plurality of BPSK symbols at a time.

[0020] In accordance to an embodiment, a device includes: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a

plurality of channel; after transmitting the first synchronization sequence, transmitting, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data; monitor, using the transceiver, a single second synchronization channel for detection of a second synchronization sequence associated with a second device, and monitor, using the transceiver, the single second synchronization channel for detection of a third synchronization sequence associated with a third device.

[0021] In accordance to an embodiment, a device including: a transceiver; and a controller configured to: generate a first plurality of spreads of chips corresponding to a plurality of bits of a first packet; generate a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; transmit, via the transceiver, the first plurality of pairs of BPSK symbols in respective channels of a plurality of channels according to a first hopping sequence; generate a second plurality of spreads of chips corresponding to a plurality of bits of a second packet; generate a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; and transmit, via the transceiver, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a second hopping sequence, where the second hopping sequence is different from the first hopping sequence, and where only a single channel of the plurality of channels is used for transmission at a time.

[0022] In accordance to an embodiment, a device including: a transceiver; and a controller configured to: negotiate, via the transceiver, with a second device a first symbol duration; transmit, via the transceiver, a first synchronization sequence in a single synchronization channel of a plurality of channels using the negotiated first symbol duration; and after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time using the negotiated symbol duration.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] For a more complete understanding of the present disclosure, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

[0024] FIG. 1 illustrates various exemplary channel arrangements for wireless communication;

[0025] FIG. 2 illustrates a communication system, according to an embodiment of the present disclosure;

[0026] FIG. 3 shows a schematic diagram illustrating an RF core, according to an embodiment of the present disclosure;

[0027] FIG. 4 shows a block diagram of a processing pipeline for generating a modulated signal for wireless transmission, according to an embodiment of the present disclosure;

[0028] FIGS. 5 and 6 show block diagrams of processing pipelines for processing a received modulated signal, according to embodiments of the present disclosure;

[0029] FIG. 7 shows a schematic diagram of a scrambler, according to an embodiment of the present disclosure;

[0030] FIG. 8 shows a schematic diagram of a de-scrambler, according to an embodiment of the present disclosure;

[0031] FIG. 9 shows a block diagram of a forward error correction (FEC) encoder, according to an embodiment of the present disclosure;

[0032] FIG. 10 shows a schematic diagram of a convolutional encoder, according to an embodiment of the present disclosure;

[0033] FIG. 11 shows a block diagram illustrating the operation of an interleaver, according to an embodiment of the present disclosure;

[0034] FIG. 12 shows a block diagram illustrating the operation of a de-interleaver, according to an embodiment of the present disclosure;

[0035] FIG. 13 shows a block diagram of a direct sequence spread spectrum (DSSS) modulator, according to an embodiment of the present disclosure;

[0036] FIGS. 14 and 15 show chip sequences for different DSSS values and polarities, according to embodiments of the present disclosure;

[0037] FIG. 16 shows a block diagram of a DSSS demodulator, according to an embodiment of the present disclosure;

[0038] FIG. 17 shows a block diagram of a BPSK encoder, according to an embodiment of the present disclosure;

[0039] FIG. 18 shows a block diagram of a BPSK decoder, according to an embodiment of the present disclosure;

[0040] FIG. 19 shows a block diagram of a single sub-carrier mapper, according to an embodiment of the present disclosure;

[0041] FIG. 20 shows a block diagram of an Inverse Fast Fourier Transform (IFFT) block 2000, according to an embodiment of the present disclosure;

[0042] FIG. 21 shows a block diagram of a Fast Fourier Transform (FFT) block, according to an embodiment of the present disclosure;

[0043] FIG. 22 illustrates the operation of a cyclic prefix with windowing function, according to an embodiment of the present disclosure;

[0044] FIGS. 23A-23D show data rates for various possible settings of an RF core, according to an embodiment of the present disclosure;

[0045] FIG. 24 shows a flow chart of an embodiment method for selecting communication parameters, according to an embodiment of the present disclosure;

[0046] FIG. 25 illustrates a packet structure of a physical layer (PHY) protocol data unit (PPDU), according to an embodiment of the present invention;

[0047] FIG. 26, shows a short training field (STF) bit sequence, according to an embodiment of the present disclosure;

[0048] FIG. 27, shows a long training field (LTF) bit sequence, according to an embodiment of the present disclosure;

[0049] FIG. 28 illustrates transmission of a packet encoded across multiple sub-carrier frequencies, according to an embodiment of the present disclosure;

[0050] FIGS. 29A and 29B show sets of possible values for two coefficients, respectively, of a linear congruential generator (LCG) for determining a hopping sequence, according to an embodiment of the present disclosure;

[0051] FIGS. 30-33 shows transmission of a plurality of packets, according to an embodiment of the present disclosure;

[0052] FIG. 34 shows a block diagram of a modulator, according to an embodiment of the present disclosure;

[0053] FIGS. 35 and 36 show block diagrams of demodulators, according to embodiments of the present disclosure; [0054] FIG. 37 shows a flow chart of an embodiment method for receiving a packet, according to an embodiment of the present disclosure; [0055] FIG. 38 shows a flow chart of an embodiment method for generating a hopping sequence, according to an embodiment of the present disclosure; [0056] FIGS. 39 and 40 show flow charts of embodiment methods for packet exchanges, according to embodiments of the present disclosure; [0057] FIG. 41 illustrates a communication system, according to an embodiment of the present disclosure; [0058] FIGS. 42 and 43 shows a flow chart of an embodiment method for packet exchange, according to an embodiment of the present disclosure; [0059] FIG. 44 illustrates symbol transmission of a communication system, according to an embodiment of the present disclosure; and [0060] FIGS. 45 and 46 illustrate communication systems, according to embodiments of the present disclosure.

[0061] Corresponding numerals and symbols in different figures generally refer to corresponding parts unless otherwise indicated. The figures are drawn to clearly illustrate relevant aspects of preferred embodiments and are not necessarily drawn to scale.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0062] The making and using of the embodiments disclosed are discussed in detail below. It should be appreciated, however, that the present disclosure provides many applicable inventive concepts that can be embodied in a wide variety of specific contexts. The specific embodiments discussed are merely illustrative of specific ways to make and use the disclosure, and do not limit the scope of the disclosure.

[0063] The description below illustrates various specific details to provide an in-depth understanding of several example embodiments according to the description. The embodiments may be obtained without one or more of the specific details, or with other methods, components, materials and the like. In some cases, known structures, materials or operations are not shown or described in detail so as not to obscure the different aspects of the embodiments. References to “an embodiment” in this description indicate that a particular configuration, structure or feature described in relation to the embodiment is included in at least one embodiment. Consequently, phrases such as “in one embodiment” that may appear at different points of the present description do not necessarily refer exactly to the same embodiment. Furthermore, specific formations, structures or features may be combined in any appropriate manner in one or more embodiments.

[0064] Several aspects of the disclosure are described below with reference to example applications for illustration. It should be understood that numerous specific details, relationships, and methods are set forth to provide an understanding of the disclosure. The present disclosure is not limited by the illustrated ordering of acts or events, as some acts may occur in different orders and/or concurrently with other acts or events.

[0065] Embodiments of the present disclosure are described in specific contexts, e.g., a long-range OFDM-

based wireless communication system and method, e.g., suitable for internet-of-things (IoT) devices. In some embodiments, long-range OFDM modulation offers a good tradeoff between long range (e.g., >150 dB of link budget), network capacity (e.g., multiple code access), and being a standardized solution without expensive central nodes and carrier subscriptions of other protocols. Long-range OFDM can also allow for good utilization of the time/frequency grid. Some embodiments may be used in short-range wireless communication systems. Some embodiments may not be OFDM-based, and may rely in other schemes, such as non-orthogonal FDM.

[0066] Some embodiments may operate in sub-1 GHz band(s) (e.g., bands between 470 MHz and 925 MHz). Some embodiments may, alternatively or in addition to sub-1 GHz band(s), operate in bands above 1 GHz, such as 2.4 GHz, 5 GHz, 6 GHz, 7 GHz, or higher, such as 60 GHz or higher.

[0067] Some embodiments may be implemented in or for IoT devices, such as in or for sensor devices that collect data and transmit such sensed data and/or for devices used for (e.g., remotely) controlling another device. In some embodiments, such IoT devices are battery powered (and may not be powered by mains). In some embodiments, such IoT devices may be battery-less (e.g., implemented with a small battery) or battery-free (implemented without a battery) and may harvest energy using energy harvesting methods, such as backscattering.

[0068] Some embodiments may be implemented in or for devices that may not be considered IoT devices.

[0069] Some embodiments may be implemented in or for devices powered by mains.

[0070] Some embodiments may be used in applications such as asset management, such as applications for monitoring/tracking assets. For example, in some embodiments, a device (e.g., an IoT device) may be attached to an asset (e.g., a tool, package, truck, etc.) and transmit location and/or other information/data to a receiver.

[0071] Some embodiments may be used in applications such as agriculture, such as applications for monitoring/tracking cattle, soil conditions etc. For example, in some embodiments, a device may be attached to cattle and transmit location and/or other information/data (e.g., health status, etc.) to a receiver. As another example, in some embodiments, a device may sense soil conditions (e.g., humidity, etc.) and transmit such sensed data to a receiver.

[0072] Some embodiments may be used in applications such as smart city. For example, in some embodiments, a device may monitor/track status of parking spots, and may transmit such information/data to a receiver (e.g., to allow for a driver to find an empty parking spot). As another example, in some embodiments, a device may be used to receive a controlling signal and control street lights based on the received signal.

[0073] Some embodiments may be used in metering applications. For example, in some embodiments, a device may monitor/track one or more parameters associated to electricity, water, and/or gas usage. The device may then transmit sensed data to a receiver.

[0074] In an embodiment of the present disclosure, a seed value for selecting a hopping sequence for wireless data transmission (e.g., for transmission of a long training field, a header field and/or a payload field of a packet) is selected based on a synchronization sequence (e.g., a short training field) associated with the packet.

[0075] In some embodiments, the seed value is selected based on which sub-channel the synchronization sequence is transmitted. In some embodiments, the seed value is selected based on one or more bits of the synchronization sequence.

[0076] In some embodiments, the hopping sequence is determined based on the seed value and on one or more coefficients. In some such embodiments, the seed value and/or the one or more coefficients change for each packet.

[0077] In some embodiments, the use of different hopping sequences may advantageously reduce the probability of collision between different transmitters, which may advantageously allow for multiple independent networks to coexist and use the same available sub-channels in the same geographical area, e.g., for long range transmissions. The use of different hopping sequences may also advantageously allow for a transmitter to (e.g., independently) serve multiple receivers (e.g., using a different synchronization sequence, synchronization channel, and/or hopping sequence for each).

[0078] In general, frequency-division multiplexing (FDM) transmitters encode information in multiple frequency bands and combine signals from the frequency bands for transmission. FDM systems have high throughput, as compared to some other communication systems. However, a transmitter implementing FDM may consume large amounts of power at peak conditions, resulting in a high peak-to-average-power ratio (PAPR). For example, the SUN OFDM PHY described in chapter 20 of IEEE Std 802.15.4-2020 and incorporated herein by reference, implements an OFDM modulation scheme that may exhibit a PAPR of about 8-9 dB.

[0079] In an embodiment, a transmitter using channels of an OFDM-based channel arrangement (e.g., having sub-channels overlapping in frequency), may encode information/data in only one sub-channel at a time, which may result in reduced PAPR (e.g., PAPR closer to one) compared to conventional OFDM systems. A reduced PAPR may advantageously result in higher data throughput (e.g., transmission of more bits per second) compared to systems with higher PAPR. In some embodiments, a reduced PAPR may be advantageous for long range transmissions.

[0080] Interference or noise experienced by a transmitter, receiver, or transceiver may be classified as vertical interference and horizontal interference. In an example of vertical interference, a transmitter, receiver, or transceiver, experiences a short burst of interference that impacts all of the frequencies used by transmitter, receiver, or transceiver. In an example of horizontal interference, a transmitter, receiver, or transceiver, experiences narrowband interference that impacts some, but not all, of the frequency bands in which the transmitter, receiver, or transceiver, operates.

[0081] In some embodiments, a transmitter may spread information/data across time using techniques such as direct sequence spread spectrum (DSSS). By spreading information/data across time, some embodiments may be advantageously more robust and resilient to vertical interference, such as temporary burst of energy that may temporarily jam or otherwise render unusable one or more (or all) communication channels available to the transmitter. In addition, DSSS spreading may advantageously provide redundancy for low sensitivity in some embodiments. As such, in some embodiments, transmitting information/data spread across time may advantageously facilitate long range transmissions.

[0082] Some embodiments may use, instead or in addition to DSSS, other spread spectrum techniques, such as frequency-hopping spread spectrum (FHSS), time-hopping spread spectrum (THSS), chirp spread spectrum (CSS), and/or a combination of two or more of DSSS, FHSS, THSS, and CSS.

[0083] In some embodiments, a transmitter may spread information across multiple frequency bands using techniques such as single carrier-frequency-division multiple access (SC-FDMA). By spreading information/data across frequency, some embodiments may be advantageously more robust and resilient to horizontal interference, such as temporal or permanent unavailability of one or more communication channels available to the transmitter (e.g., due to jamming, noise or other factors that may render a communication channel unusable). In some embodiments, a single carrier OFDM has no back-off and a PAPR of zero dB, creates an orthogonal time/frequency grid, and/or uses a set of pseudo-random codes that provide code diversity between nodes and networks. As such, in some embodiments, transmitting information/data spread across frequency may advantageously facilitate long range transmissions.

[0084] In some embodiments, a transmitter may use error correction techniques, such as forward error correction (FEC), which may advantageously allow a receiver to recover corrupted data. By enabling error detection and correction, in some embodiments, a receiver may be able to reconstruct the received data, even when portions of the received data is incomplete (e.g., due to lost packets, e.g., as a result of one or more (or all) channels being temporary unavailable, and/or due to one or more channels being permanently unavailable). As such, in some embodiments, transmitting information/data with error correction capabilities may advantageously facilitate long range transmissions.

[0085] In some embodiments, a transmitter may use cyclic prefix to repeat all (or a portion) of each symbol (e.g., before or after transmission of each symbol), which may advantageously result in additional transmission redundancy. As such, in some embodiments, transmitting information/data with cyclic prefix may advantageously facilitate long range transmissions.

[0086] In some embodiments, a transmitter may add a guard interval between symbols, which may advantageously reduce intersymbol interference between adjacent symbols. As such, in some embodiments, transmitting symbols with an intersymbol guard interval may advantageously facilitate long range transmissions.

[0087] In some embodiments, a transmitter may transmit symbols in pairs, where each pair of symbols is transmitted in a single sub-channel, and where each symbol of each pair is differentially encoded (e.g., using binary phase shift keying (BPSK)). By transmitting a pair of differentially encoded symbols (e.g., using BPSK) in a single sub-channel, in some embodiments, the pair of symbols share a common phase reference, which may advantageously allow for successful decoding without precise channel equalization. Further, in some embodiments, BPSK may provide a good demodulation signal-noise ratio.

[0088] Some embodiments may use, instead or in addition to BPSK, other digital modulation techniques, such as frequency-shift keying (FSK), gaussian FSK (GFSK), amplitude-shift keying (ASK), quadrature amplitude modulation (QAM), amplitude and phase-shift keying (APSK), continuous phase modulation (CPS), minimum shift keying

(MSK), on-off keying (OOK), and/or a combination of two or more of PSK, GFSK, BPSK, FSK, ASK, QAM, APSK, CPS, MSK and OOK.

[0089] In some embodiments, each pair of symbols is sent in respective sub-channels selected in accordance with a hopping sequence. In some embodiments, the use of different hopping sequence may reduce the probability of collision between different transmitters operating according to different hopping sequences, which may advantageously allow for multiple independent networks to coexist and use the same available sub-channels in the same geographical area, e.g., for long range transmissions.

[0090] In some embodiments, a synchronization sequence is used to transmit a seed indicative of the hopping sequence from a transmitter to a receiver. As such, in some embodiments, a transmitter may independently serve a plurality of receivers. In some embodiments, multiple transmitter may independently serve multiple receivers by each transmitter using a distinct hopping sequence, e.g., for long range transmissions.

[0091] In some embodiments, the synchronization sequence is transmitted (e.g., in its entirety) in a single (e.g., predetermined) sub-channel. As such, in some embodiments, a (e.g., low power) receiver (e.g., such as an IoT device) may periodically transition from a low power mode to a high power mode to listen to a predetermined sub-channel. Upon detection of the synchronization sequence, the receiver may begin hopping according to a hopping sequence based on the synchronization sequence for transmission and/or reception of data. As such, in some embodiments, using a single predetermined sub-channel for transmission of the synchronization sequence may advantageously allow for low power consumption of a receiver (e.g., since the receiver monitors a single sub-channel for predetermined windows of times as opposed to scanning multiple sub-channels).

[0092] In some embodiments, multiple independent networks may advantageously coexist and use the same available sub-channels in the same geographical area by transmitting respective synchronization sequences in respective (different) sub-channels.

[0093] FIG. 2 illustrates communication system 200, according to an embodiment of the present disclosure. Communication system 200 includes wireless devices 202 and 252 communicating via network 210.

[0094] In some embodiments, communication between device 202 and device 252 may be symmetrical. As such, in some embodiments, the method for transmitting data from device 202 to device 252, and receiving data by device 252 from device 202, may be similar or identical to the method for transmitting data from device 252 to device 202, and receiving data by device 202 from device 252. Although some features of some embodiments may be described with respect to a particular device (e.g., 202/252), such features may be equally applicable to other devices (e.g., 252/202).

[0095] In some embodiments, device 202 may be an access point (AP) with access, via a wired or wireless protocol, to another network (e.g., an intranet, the Internet, etc.), while device 252 may be an IoT device, such as an IoT sensor. In some embodiments, devices 202 and 252 may each be an IoT device, such as an IoT sensor. In some embodiments, devices 202 and 252 may each be an AP device with access, via a wired or wireless protocol, to another network (e.g., an intranet, the Internet, etc.).

[0096] For clarity purposes, many of the embodiments describe assume that device 202 is an AP device while device 252 is an IoT sensor device. However, it is understood that the illustrated features and inventive concepts may equally apply to embodiments in which device 202 is not an AP (e.g., is an IoT device, or a non-IoT device that is not an AP), and/or device 202 is not an IoT sensor device (e.g., is a non-sensor IoT device, an AP device, or another device that is not considered an AP device or an IoT device).

[0097] In some embodiments, device 252 may enter a low power mode between (e.g., periodic) data transmissions, and may operate in a high power mode during the data transmissions. In some embodiments, device 252 may not enter low power mode between data transmissions.

[0098] In some embodiments, device 252 may use Pure ALOHA medium access schemes or similar while device 202 is in an always listening mode (e.g., does not enter low power mode). In some such embodiments, device 252 may transmit a packet to device 202 and then immediate change to receiver mode to receive an acknowledgement (ACK) from device 202. As a result, in some embodiments, device 252 can be in idle/low power mode the majority of the time and only spend energy when device 252 needs to transmit data and then receive the corresponding ACK. This mechanism may be understood as half-duplex.

[0099] In some embodiments, device 202 may enter a low power mode between (e.g., periodic) data transmissions, and may operate at a high power mode during the data transmissions. In some embodiments, device 202 may not enter low power mode between transmissions.

[0100] In some embodiments, device 252 may send one or more packets (e.g., including sensor data, and/or a command) at any time (e.g., synchronously or asynchronously) to device 202; device 202 may receive such packet(s). In response, device 202 may transmit one or more packets (e.g., including an ACK, a command, and/or data) to device 252, which may be listening for such packet(s). In some such embodiments, device 252 may enter low power mode between transmissions and device 202 may not enter low power mode between transmissions. In other such embodiments, one of devices 202 and 252 may enter low power mode between transmissions while the other one of devices 202 and 252 does not enter low power mode between transmissions. In yet other such embodiments, devices 202 and 252 do not enter low power mode between transmissions.

[0101] In some embodiments, device 202 may transmit (e.g., synchronously or asynchronously) one or more packets to device 252, e.g., not in response to packets received from device 252. In some such embodiments, device 252 may periodically wake up from low power mode to listen for communications, e.g., from device 202. In other such embodiments, device 252 may not enter low power mode between transmissions and may continuously listen for such packet(s).

[0102] During normal operation, device 202 wirelessly transmits via RF core 204 and antenna 208 first data to device 252. Device 252 receives the first data, and then decodes and processes the first data. Device 252 may then wirelessly transmit, via RF core 254 and antenna 258, second data to device 202 (or to another device, not shown in FIG. 2).

[0103] In some embodiments, devices 202 and 252 may be separated by a distance that is relatively large, such as 1 Km,

2 Km, 10 Km, 15 Km, or larger. In some embodiments, devices 202 and 252 may be separated by a distance shorter than 1 Km, such as 500 m, 100 m, 30 m, 10 m, 5 m, 1 m, or shorter.

[0104] In some embodiments, device 202 and/or device 252 may be a mobile device, and may move during transmission of data (such that the distance between devices 202 and 252 may dynamically change). In some embodiments, device 202 and/or device 252 may be at a fixed location during transmission of data.

[0105] In some embodiments, device 252 may comprise a sensor such as a temperature sensor, humidity sensor, location sensor, vibration sensor, etc. In some such embodiments, device 252 may wireless transmit data to device 202 based on sensed data sensed by the sensor.

[0106] In some embodiments, device 202 may, in response to data received from device 202, operate another device (not shown in FIG. 2) or cause an action in the another device based on the received data. In some embodiments, the another device includes a motor, a speaker, a microphone, a solenoid, a (e.g., LED) light, a solar cell, a battery, a radar, a memory, and/or other electronic circuit(s), such as a processor, power management circuits, etc. In some embodiments, device 252 may, in response to data received from device 202, cause the start, stop, or change a mode of energy harvesting; activate, deactivate, or change operation of a motor, solenoid or light; store/erase/modify data to/from a memory; play or stop playing a sound; begin or stop recording data using a sensor, such as microphone, humidity sensor, temperature sensor, etc.; activate, deactivate, or change operation of a radar; change operation of an electronic circuit of another device coupled to device 252, etc.

[0107] As shown in FIG. 2, device 202 includes RF core 204 and controller 206; and device 252 includes RF core 254 and controller 256.

[0108] In some embodiments, RF cores 204 implements the physical (PHY) layer and at least part (or all) of the data link layer (e.g., MAC layer) of wireless protocol 212 used for communicating via network 210, while controller 206 implements higher level layers (e.g., network layer, transport layer, session layer, presentation layer, and/or application layer). In some embodiments, controller 206 may implement all of the data link layer of wireless protocol 212. In some embodiments, controller 206 may implement at least part of the PHY layer of wireless protocol 212. Other implementations are also possible.

[0109] In some embodiments, RF core 204 is configured to assemble bits in a given packet structure for transmission using antenna 208.

[0110] In some embodiments, RF core 204 supports multiple modulation formats, including (e.g., multilevel) GFSK and MSK, OOK, BPSK, and CSS, among others.

[0111] In some embodiments, RF core 204 has dedicated handling accelerators, e.g., for forward error correction, data whitening, and/or automatic cyclic redundancy checks (CRC). In some embodiments, RF core 204 includes additional accelerators. In some embodiments, RF core 204 does not include any hardware accelerators.

[0112] In some embodiments, RF core 204 includes a wireless transceiver including a transmission path and a reception path coupled to antenna 208. In some embodiments, the transmission path includes a power amplifier having an output coupled to antenna 208. In some embodiments, the reception path includes a low-noise amplifier

having an input coupled to antenna 208. In some embodiments, RF core 204 includes one or more analog-to-digital converters (ADCs), one or more digital-to-analog converters (DACs), one or more mixers, a combiner circuit (e.g., a parallel to serial circuit), and/or a splitter circuit (e.g., a serial to parallel circuit), a (e.g., digital) phase-locked-loop (PLL), a modem, read-only memory (ROM), random-access memory (RAM), such as SRAM, one or more filters, and/or one or more amplifiers, to facilitate wireless transmission and reception of data using antenna 208.

[0113] In some embodiments, such as shown in FIG. 2, a single antenna 208 is used for transmission and reception of data. In some embodiments, more than one antenna may be used for transmission and reception of data (e.g., one or more antennas may be used for transmission while other antenna(s) may be used for reception).

[0114] In some embodiments, controller 206 may generate or cause generation of data to be wirelessly transmitted by RF core 204 via antenna 208, and/or may process data received by RF core 204 via antenna 208.

[0115] In some embodiments, controller 206 may be implemented as a generic or custom controller, processor, or processing core, e.g., coupled to a memory and configured to execute instructions in such memory. In some embodiments, controller 206 may be implemented using a field programmable gate array (FPGA). In some embodiments, controller 206 includes a state machine. Other implementations are also possible.

[0116] In some embodiments (such as shown in FIG. 2), controller 206 may be external to RF core 204. In some embodiments, controller 206 may be partially or entirely implemented inside RF core 204.

[0117] In some embodiments, RF core 204 and controller 206 may be implemented in a single monolithic semiconductor substrate. In some embodiments, RF core 204 and controller 206 may be implemented in different dies in a single package. In some embodiments, RF core 204 and controller 206 may be discrete integrated circuits implemented in a printed circuit board (PCB). Other implementations are also possible.

[0118] As shown in FIG. 2, RF core 204 may be coupled to antenna 208. In some embodiments, antenna 208 is external to a package including RF core 204. In some embodiments, antenna 208 is implemented in the same package as RF core 204.

[0119] In some embodiments, RF core 254 may be implemented in a similar or identical manner as RF core 204. In some embodiments, controller 256 may be implemented in a similar or identical manner as controller 206. In some embodiments, device 252 may be implemented in a similar or identical manner as device 202.

[0120] In some embodiments, devices 202 and 252 may be instances of a same design. As such, in some embodiments, devices 202 and 252 may operate using similar or identical configurations. In some embodiments, although the design of devices 202 and 252 may be similar or identical, devices 202 and 252 may operate with different configurations (e.g., which may be programmed into such devices in response to communications via network 210, by a manufacturer of devices 202 and/or 252, and/or by a user of the device 202 and/or 252, e.g., not via network 210).

[0121] FIG. 3 shows a schematic diagram illustrating RF core 300, according to an embodiment of the present disclosure. RF core 204 and/or 254 may be implemented as RF core 300.

[0122] RF core 300 includes PLL 306, transmitter path 310, and receiver path 320. Transmitter path 310 includes DAC 311, pre-amplifier 312, mixer 314, analog filter 316, and power amplifier 318. Receiver path 320 includes low-noise amplifier (LNA) 322, mixer 324, intermediate frequency (IF) amplifier 326, analog filter 328, and ADC 330.

[0123] In some embodiments, during normal operation, when RF core 300 is in transmit mode, transmitter path 310 receives TX modulated signal from modulator 302 for wireless transmission via antenna 308. In some embodiments, when RF core 300 is in receive mode, receiver path 320 generates RX modulated signal based on signals received from antenna 308, and provides the RX modulated signal to demodulator 304 for further processing.

[0124] In some embodiments, antenna 308 is coupled to amplifiers 318 and 322 via a duplexer (not shown).

[0125] In some embodiments, RF core 300 also includes modulator 302 and/or demodulator 304. In some embodiments, modulator 302 and/or demodulator 304 includes, or may be implemented using, a generic or custom controller, processor, or processing core, e.g., coupled to a memory and configured to execute instructions in such memory. In some embodiments, modulator 302 and/or demodulator 304 includes, or may be implemented using, a field programmable gate array (FPGA). In some embodiments, modulator 302 and/or demodulator 304 includes a state machine. In some embodiments, modulator 302 and/or demodulator 304 includes or may be implemented as a hardware accelerator (s). Other implementations are also possible.

[0126] In some embodiments, modulator 302 and/or demodulator 304 are external to RF core 300 (e.g., are part of a controller (e.g., 206, 256) external to RF core 300).

[0127] In some embodiments, analog filter 276 and/or 288 include low pass or bandpass filters, and may have programmable gain. Elements 306, 312, 314, 316, 318, 322, 324, 326, 328, and 330 may be implemented in any way known in the art.

[0128] In some embodiments, modulator 302 generates a digital signal (TX modulated signal) to be transmitted by transmitter path 310 via antenna 308. In some embodiments, modulator 302 includes a DAC to convert such digital signal into a corresponding modulated analog signal provided to the input of pre-amplifier 312. In some embodiments, such DAC is external to modulator 302 and may be part of transmitter path 310 (such as shown in FIG. 3).

[0129] In some embodiments, demodulator 304 processes received digital signals (RX modulated signal) provided by ADC 330 from receiver path 320.

[0130] FIG. 4 shows a block diagram of processing pipeline 400 for generating TX modulated signal for wireless transmission, according to an embodiment of the present disclosure. Processing pipeline 400 may be implemented by modulator 302 (e.g., partially or entirely using dedicated circuits, such as hardware accelerators; and/or partially or entirely by executing instructions stored in a memory).

[0131] Processing pipeline 400 includes scrambler block 402, FEC block 404, interleaver block 406, DSSS block 408, bits-to-symbols block 410, single sub-carrier mapper block 412, inverse transform 414, cyclic prefix block 416, and digital filter block 418. In some embodiments, one or more

of the blocks of processing pipeline 400 may be omitted or additional blocks (not shown) may be performed. In some embodiments, different portions of the packet are processed by different blocks (e.g., some portions of the packet may be scrambled while other portions may not be scrambled). Other implementations are also possible.

[0132] During normal operation, a packet 401 to be wirelessly transmitted via antenna 308 (e.g., using wireless protocol 212) may be received by processing pipeline 400. For example, in some embodiments, a payload is received (e.g., from the data link layer), and packet 401 is created (e.g., by the PHY layer), e.g., by adding a header and/or other fields to the payload.

[0133] In some embodiments, processing pipeline 400 receives (e.g., sequentially), a plurality of packets 401 to be transmitted. Processing pipeline 401 may process each of the plurality of packets 401 sequentially, in parallel, and/or in a pipelined manner.

[0134] In some embodiments, scrambler 402 scrambles bits of the packet 401 to generate a scrambled bit stream. By scrambling the bits of the packet, some embodiments may advantageously improve signal quality (e.g., by preventing long sequences of 0s or 1s, and increase security for the transmission.

[0135] In some embodiments, FEC encoder 404 modifies the scrambled bit stream to generate an FEC bit stream that allows for error detection and correction. The use of FEC may advantageously allow for the recovery of data affected by horizontal or vertical interference.

[0136] In some embodiments, interleaver block 406 interleaves bits of the FEC bit stream across time to generate an interleaved bit stream. By interleaving the bits to be transmitted, some embodiments advantageously improve robustness of data transmission over vertical interference by spreading the data over time.

[0137] In some embodiments, DSSS block 408 encodes, e.g., each bit of the interleaved bit stream, into multiple bits (also referred to as chips). By adding redundancy using DSSS, some embodiments advantageously increase resistance to horizontal and vertical interference and jamming, and improve signal reliability. For example, in some embodiments, RF core 300 (e.g., demodulator 304, e.g., processing pipelines 500 or 600) or an associated controller (e.g., 206, 256) may be able to identify a bit in a received signal (e.g., RX modulated signal) even if one of the chips associated with such bit is corrupted by interference. For example, a short burst of interference may corrupt a chip, but the receiver can evaluate adjacent chips to identify the true value of the corrupted bit.

[0138] In some embodiments, DSSS block 408 may implement other spread spectrum techniques, in addition to or as an alternative to DSSS. Additional example details of DSSS in a communication system can be found in commonly assigned U.S. Patent Application Publication No. 2022/0255580, entitled “Frequency-Division Multiplexing,” filed on Dec. 17, 2021, commonly assigned U.S. Pat. No. 9,935,681, entitled “Preamble Sequence Detection of Direct Sequence Spread Spectrum (DSSS) Signals,” issued on Apr. 3, 2018, and commonly assigned U.S. Pat. No. 9,831,909, entitled “DSSS Inverted Spreading for Smart Utility Networks,” issued on Nov. 28, 2017, each of which is incorporated by reference in its entirety.

[0139] In some embodiments, bits-to-symbols block 410 converts each chip of the TX chip stream into a (e.g.,

OFDM) symbol to generate a symbol stream. In some embodiments, each symbol may be represented as a complex number with a real value and an imaginary value.

[0140] In some embodiments, bits-to-symbols block 410 may use a modulation process such as quadrature amplitude modulation (QAM) (e.g., 16-QAM) or phase shift keying (PSK) (e.g., binary PSK or quadrature PSK). In some embodiments, the ratio of chips representing each bits to symbols may be one-to-one, one-to-two, two-to-one, four-to-one, or any other ratio. For example, in some embodiments using BPSK, bits-to-symbols block 410 may convert each chip from DSSS block 408 to a respective symbol. Additional example details of PSK and QAM can be found in commonly assigned U.S. Pat. No. 9,001,948, entitled "Pulse Shaping in a Communication System," issued on Apr. 7, 2015, which is incorporated by reference.

[0141] In some embodiments, single sub-carrier mapping block 412 generates a mapped stream in which each symbol of the symbol stream is mapped into a single sub-carrier signal (using only a single sub-channel of the available sub-channels). For example, in some embodiments, the mapped stream maps each symbol to a single input of inverse transform block 414, setting all other inputs of inverse transform block 414 to zero. Using a single sub-carrier signal at a time may advantageously result in lower PAPR and produce an RF signal that is easier for a receiver to demodulate.

[0142] Although a single sub-carrier may be used at a time, some embodiments change the sub-carrier periodically, e.g., according to a hopping sequence, which may advantageously spread the signal in frequency, which may advantageously improve robustness over horizontal interference.

[0143] In some embodiments, inverse transform block 414 computes an inverse Fast Fourier Transform (IFFT) on the symbols of the mapped stream to generate an inverse transformed stream that includes time-domain samples of the mapped symbols.

[0144] In some embodiments, cyclic prefix block 416 appends or prepends all or a portion of each symbol of the mapped stream to generate a prefixed symbol stream. Such cyclic prefix addition may advantageously act as a guard interval that may help reduce intersymbol interference, which may be an important consideration in multipath environments.

[0145] In some embodiments, digital filtering block 418 filters the samples of prefixed symbol stream to produce TX modulated signal, which may be provided to transmit path 310 for wireless transmission via antenna 308.

[0146] In some embodiments, digital filtering block 418 is not implemented and the samples generated by cyclic prefix block 416 constitute the TX modulated signal to be transmitted by antenna 308. In some embodiments, blocks 416 and 418 are not implemented and the output of inverse transform block 414 constitute the TX modulated signal to be transmitted by antenna 308. In some embodiments cyclic prefix block 416 is not implemented but digital filtering block 418 is implemented.

[0147] In some embodiments, one or more (or all) of blocks 402, 404, 406, 408, 410, 412, 414, 416, and 418 is configurable. For example, in some embodiments, one or more (or all) of blocks 402, 404, 406, 408, 410, 412, 414, 416, and 418 is configurable based on data received during a prior communication between devices 202 and 252.

[0148] In some embodiments, the TX modulated signal is wirelessly transmitted by transmitter path 310 of device 202, and is received by receiver path 320 of device 252 as RX modulated signal.

[0149] FIG. 5 shows a block diagram of processing pipeline 500 for processing RX modulated signal, according to an embodiment of the present disclosure. Processing pipeline 500 may be implemented by demodulator 304 (e.g., partially or entirely using dedicated circuits, such as hardware accelerators; and/or partially or entirely by executing instructions stored in a memory).

[0150] Processing pipeline 500 includes digital filtering block 502, Fourier transform block 504, extraction from carrier block 506, inverse DSSS block 508, de-interleaver block 510, symbols-to-bits metrics block 512, FEC decoder 514, and de-scrambler block 516. In some embodiments, one or more of the blocks of processing pipeline 500 may be omitted or additional blocks (not shown) may be performed. In some embodiments, different portions of the RX modulated signal are processed by different blocks (e.g., some portions of the signal may be processed by inverse DSSS block 508 while other portions may not be processed by inverse DSSS block 508). Other implementations are also possible.

[0151] During normal operation, an RX modulated signal wirelessly may be received via antenna 308 may be received by processing pipeline 500. In some embodiments, digital filtering block 502 filters the RX modulated signal (e.g., using low pass filtering) to generate a filtered signal.

[0152] In some embodiments, Fourier transform block 504 performs, e.g., after cyclic prefix removal, a Fourier transform (e.g., DFT, such as FFT), e.g., in an inverse manner than inverse transform block 414 so as to recover the symbols generated by single sub-carrier mapping block 412.

[0153] In some embodiments, extraction from carrier block 506 processes the Fourier transformed stream generated by Fourier transform block 504 to determine the single carrier frequency containing the symbols and generates an RX chip symbol stream with symbols based, e.g., on the phase, frequency, and/or modulation detected on the carrier signal, e.g., in accordance with a hopping sequence associated with the RX modulated signal.

[0154] In some embodiments, inverse DSSS block 508 uses the redundancy introduced by DSSS block 408 to generate a symbol metric stream, where each symbol metric of the symbol metric stream is indicative of the likelihood of a symbol corresponding to the associated multiple chips.

[0155] In some embodiments, de-interleaver block 510 de-interleaves the symbol metric stream (e.g., in an inverse manner as interleaver block 406) to produce a de-interleaved symbol metric stream.

[0156] In some embodiments, the symbols-to-bits metrics block 512 converts the de-interleaved symbol metric stream to a bit metric stream, (e.g., in an inverse manner as bits-to-symbols block 410), where each bit metric of the metric stream is indicative of the likelihood of a bit received.

[0157] In some embodiments, FEC decoder block 514 performs error correction on the bit metric stream so as to identify errors and correct any correctable errors, to generate an error-corrected bit stream.

[0158] In some embodiments, de-scrambler block 516 descrambles the error-correct bit stream (e.g., in an inverse manner as scrambler block 402) to generate a reconstructed packet 501.

[0159] In some embodiments, the reconstructed packet **501**, when reconstructed successfully, may be identical to the packet **401**. In some embodiments, the reconstructed packet **501**, may include some, but not all fields of packet **401**. For example, in some embodiments, a synchronization field (e.g., a short training field) that may be part of packet **401** may not be part of reconstructed packet **501**. In some embodiments, the reconstructed packet **501** may include only a payload (e.g., identical to the payload of packet **401**), while omitting all other fields.

[0160] In some embodiments, processing pipeline **500** processes RX modulated signal (e.g., continuously, e.g., for a predetermined period of time) so that when X packets **401** are transmitted (e.g., by device **202**), X reconstructed packets are produced by processing pipeline **500** (e.g., of device **252**).

[0161] In some embodiments, one or more (or all) of blocks **502**, **504**, **506**, **508**, **510**, **512**, **514**, and **516** is configurable. For example, in some embodiments, one or more (or all) of blocks **502**, **504**, **506**, **508**, **510**, **512**, **514**, and **516** is configurable based on data received during a prior communication between devices **202** and **252**.

[0162] Some embodiments, such as shown in FIG. 5, may perform the inverse DSSS and de-interleaving steps on metrics, which may advantageously allow for improved error rate, which may be due to the RX chip symbol stream exhibiting noise and other artifacts (e.g., symbols may have crossed boundaries of other symbols). Other implementations are also possible. For example, FIG. 6 shows a block diagram of processing pipeline **600** for processing RX modulated signal, according to an embodiment of the present disclosure. Processing pipeline **600** may be implemented by demodulator **304** (e.g., partially or entirely using dedicated circuits, such as hardware accelerators; and/or partially or entirely by executing instructions stored in a memory).

[0163] Processing pipeline **600** operates in a similar manner as processing pipeline **500**, and may generate a reconstructed packet **501** that may be identical to the reconstructed packet **501** generated by processing pipeline **500**. Processing pipeline **600**, however, performs a symbols-to-bits conversion (block **602**) prior to performing the inverse DSSS (block **604**) and de-interleaving steps (block **606**). Other implementations are also possible.

[0164] FIG. 7 shows a schematic diagram of scrambler **700**, according to an embodiment of the present disclosure. Scrambler block **402** may be implemented as scrambler **700**. Scrambler **700** includes pseudo-noise (PN) generator **720**, and XOR gate **730**. PN generator **720** includes a plurality of flip-flops (FFs) **722**, and XOR gate **724**. In some embodiments, PN generator **720** is loaded/initialized with a seed (e.g., all ones—11111111), and is clocked using the seed as the starting point and enabled after the first clock cycle.

[0165] During normal operation, XOR gate **730** receives an input bit stream (e.g., of bits from packet **401**), and an output PN_{out} from PN generator **720** to generate a scrambled bit stream.

[0166] As can be seen in FIG. 7, In some embodiments, the scrambled bits are found using XOR operation of each of the input bits with the output (PN_{out}) of the PN generator, e.g., as follows:

$$\text{bit}_n = (\text{input bit}_n) \text{ } XOR \text{ } (PN_{out}) \quad (1)$$

[0167] FIG. 8 shows a schematic diagram of de-scrambler **800**, according to an embodiment of the present disclosure. De-scrambler block **516** may be implemented as de-scrambler **800**.

[0168] In some embodiments, de-scrambler **800** may be implemented in an inverse manner as scrambler **700**. For example, as shown in FIG. 8, in some embodiments, de-scrambler **800** includes the same PN generator **720** (initialized in the same manner) as scrambler **700**, and provides the PN_{out} of such generator **720** to XOR gate **802**, where the other input of XOR gate receives the scrambled bit stream, and the output of XOR gate **802** produces the de-scrambled bit stream.

[0169] FIG. 9 shows a block diagram of FEC encoder **900**, according to an embodiment of the present disclosure. FEC encoder block **404** may be implemented as FEC encoder **900**. FEC encoder **900** includes encoder **902**.

[0170] During normal operation, encoder **902** performs forward error correction by performing convolutional coding, channel coding, and/or polar coding on an input bit stream (e.g., from XOR gate **730**) to generate the FEC bit stream. In some embodiments, encoder **902** uses a concatenated code including a Reed-Solomon block code and an inner half-rate convolutional code.

[0171] In some embodiments, FEC encoder **900** receives the input bits stream as a sequence of M bits, where M is a multiple of 8. In some embodiments, M may have a different value, such as higher than 8, e.g., 16, 32, or higher, or lower than 8, e.g., 4 or lower.

[0172] FIG. 10 shows a block diagram of convolutional encoder **1000**, according to an embodiment of the present disclosure. Encoder **902** may be implemented as convolutional encoder **1000**.

[0173] As shown in FIG. 10, convolutional encoder **1000** may include two outputs (Output Data A and Output Data B). In some embodiments, the two outputs are subsequently serialized to form a single FEC bit stream, which may be provided to a subsequent processing block (e.g., interleaver **406**).

[0174] In some embodiments (e.g., as shown in FIG. 10), the input bits received by convolutional encoder **1000** are coded with a convolutional encoder of coding rate R=½. In some embodiments, the convolutional encoder uses the generator polynomials expressed in octal representation g₀=133₈ and g₁=171₈. Other implementations are also possible. For example, in some embodiments, the coding rate R may be different than ½, such as ¾ or different, and/or may use different polynomials.

[0175] In some embodiments, convolutional encoder **1000** is initialized to the all zeros state before encoding the input bits and then reset to the all zeros states.

[0176] In some embodiments, FEC decoder **514** may be implemented in any way known in the art, so as to perform error correction on a bit stream generated using FEC encoder **900** (e.g., implementing encoder **1000**).

[0177] FIG. 11 shows a block diagram illustrating the operation of interleaver **1100**, according to an embodiment of the present disclosure. Interleaver block **406** may be implemented as interleaver **1100**.

[0178] As can be seen in FIG. 11, in some embodiments, interleaver **1100** receives an input bit stream and produces an interleaved bit stream. In the example of FIG. 11, the input bit stream and interleaved bit stream each contain 16 bits (b_0 to b_{15}), where interleaver **1100** rearranges the bits (e.g., b_0 to b_{15}) so that the interleaved bit stream contains the same bits as input bit stream, but in a different order.

[0179] In some embodiments, interleaver **1100** operates in a similar manner on symbols instead of bits. Thus, in some embodiments, interleaver **1100** receives an input symbol stream and produces an interleaved symbol stream, e.g., in a similar manner as described with respect to bits.

[0180] In some embodiments, interleaver **1100** takes as input a sequence of Q coded bits (e.g., coded by FEC encoder **900**) and produces a second sequence of Q interleaved bits, where Q is a positive integer, such as a positive integer multiple of M. In some embodiments (as can be seen in FIG. 11), Q is equal to 16. In some such embodiments, M is equal to 8.

[0181] In some embodiments, the complete sequence of coded bits of length N produced by interleaver **1100** is defined as $C = \{c(i)\}, 0 \leq i \leq N-1$. In some embodiments, N is a multiple of 16 when M is equal to 8.

[0182] In some embodiments, a collection of consecutive subsequences of N bits produced by FEC encoder **900** may be processed by interleaver **1100** as first come first serve. In some embodiments, the collection of consecutive subsequences may be processed in a different order.

[0183] In some embodiments, interleaver **1100** performs the interleaving processes by performing 1 permutation. For example, in some embodiments, such as shown in FIG. 11, the index of the coded bit after the first permutation may be given by:

$$\text{index} = 4 \times [k \bmod F] + \text{floor}\left(\frac{k}{F}\right) \quad (2)$$

where k represents the coded bit before the first permutation and F represents a factor, mod represents the modulus operand, and floor() represents the floor function. In some embodiments, F is lower than M. In some embodiments, (such as shown in FIG. 11) F is equal to 4. In some embodiments, the interleaving process may be performed in a different manner, such as by performing more than 1 permutation, or according to a different formula.

[0184] In some embodiments, interleaver **1100** (e.g., continuously) processes the output of a preceding block (FEC encoder **404**) in groups of Q bits, e.g., as the groups of Q bits are generated.

[0185] FIG. 12 shows a block diagram illustrating the operation of interleaver **1200**, according to an embodiment of the present disclosure. De-interleaver blocks **510** or **606** may be implemented as de-interleaver **1200**.

[0186] As shown in FIG. 12, in some embodiments, de-interleaver **1200** operates in an inverse manner as interleaver **1100**, so as to remove the interleaving effect (de-interleave) introduced by interleaver **1100**.

[0187] FIG. 13 shows a block diagram of DSSS modulator **1300**, according to an embodiment of the present disclosure. DSSS block **408** may be implemented as DSSS modulator **1300**.

[0188] In some embodiments, DSSS modulator **1300** receives an input bit stream, and generates a sequence of bits

based on each bit of the input bit stream. Each of the bits of the generated sequence of bits may be referred to as a chip.

[0189] The number of chips generated for each input bit may be controlled by a DSSS value. For example, a DSSS value of 2 may cause the generation of 2 chips per input bit. Similarly, a DSSS value of 4 may cause the generation of 4 chips per input bit.

[0190] In some embodiments, the DSSS value may be a power of 2 (e.g., 2, 4, 8, 16, 32, 64, etc.). In some embodiments, the DSSS value may not be a power of 2 (e.g., 6, 10, 24, etc.). In some embodiments, the DSSS value may be an even number.

[0191] In some embodiments, the DSSS value of DSSS modulator **1300** is programmable, and may dynamically change, e.g., based on the portion of a packet (e.g., **401**) being processed.

[0192] In some embodiments, the DSSS value is selectable from a set of DSSS values that are a power of 2 (e.g., 2, 4, 8, 16). In some embodiments, the DSSS value is selectable from a set of DSSS value that may include values that are, and are not a power of 2 (e.g., 2, 4, 6, 8, 12; 2, 4, and 6; or 2, 4, 8, 12, for example). In some embodiments, the DSSS value is selectable from a set of DSSS value that only include values that are not a power of 2 (e.g., 6, 12).

[0193] In some embodiments, the DSSS value of DSSS modulator **1300** is fixed. For example, in some embodiments, the DSSS value may be a predetermined value that is built into a communication device (e.g., **202** or **252**). In some embodiments, a user may be able to set the DSSS value, such that the communication device selects a (e.g., fixed) DSSS value on user input.

[0194] The sequence of chips generated for each bit may be associated with a particular polarity. For example, when DSSS modulator **1300** operates with a DSSS value of 2 and an even polarity, an input bit of 1 may be represented by the sequence of chips [0 0], and an input bit of 0 may be represented by the sequence of chips [0 1]. When DSSS modulator **1300** operates with a DSSS value of 2 and an odd polarity, an input bit of 1 may be represented by the sequence of chips [1 1], and an input bit of 0 may be represented by the sequence of chips [1 0].

[0195] In some embodiments, the polarity of the sequence of chips is programmable, and may dynamically change, e.g., based on the portion of a packet (e.g., **401**) being processed, may toggle every predetermined number of input bits being processed (e.g., every input bit, every 2 input bits, etc.), and/or every packet (e.g., a first packet starts with an even polarity, and a subsequence packet starts with another polarity), or every predetermined number of packets. For example, DSSS modulator **1300** may alternate between even and odd spreads, may use only an even spread, or may use only an odd spread, e.g., based on which portion of packet **401** is being processed.

[0196] In some embodiments, DSSS modulator **1300** may toggle between even and odd spread, e.g., to avoid multiple repetitions of the same sequence. For example, DSSS modulator **1300** may switch between even and odd spreads after each input bit or after a particular number of input bits. For example, in some embodiments, DSSS modulator **1300** may generate a first even spread of chips representing a first input bit and then generate a second odd spread of chips representing the next input bit. Thus, in some embodiments using DSSS value equal to 2, if three consecutive input bits have a logical value of one, the transmitter can generate an even

spread of 00 representing the first input bit, an odd spread of 11 representing the second input bit, and an even spread of 00 representing the third input bit. Therefore, even though the three consecutive input bits may have the same logical value, the logical values of the chips representing the first input bit may be opposite of the logical values of the chips representing the second input bit. The logical values of the chips representing the first input bit are identical to logical values of the chips representing the third input bit but different from the logical values of the chips representing the second input bit.

[0197] In some embodiments, the polarity of the sequence of chips is fixed. For example, in some embodiments, the polarity may be a predetermined value that is built into a communication device (e.g., 202 or 252). In some embodiments, a user may be able to set the polarity, such that the communication device selects a (e.g., fixed) polarity on user input.

[0198] In some embodiments, transmitting multiple chips or symbols for each bit may advantageously improve robustness and redundancy, e.g., by spreading each bit across time and possibly across frequencies (e.g., in embodiments using frequency hopping). For example, if one chip becomes corrupted, a receiver with low sensitivity may still be able extract the value of the bit by evaluating the remaining chips in the sequence that represents the bit.

[0199] FIG. 14 shows chip sequences for different DSSS values and polarities, according to an embodiment of the present disclosure. DSSS modulator 1300 may convert input bits to chips in accordance to one or more of the tables shown in FIG. 14. In some embodiments, DSSS modulator 1300 may implement two or more of the tables shown in FIG. 14 as a selectable set (e.g., based on a selected DSSS value and/or polarity).

[0200] FIG. 14 shows a possible way of spreading input bits into sequences of chips. Other implementations are also possible. For example, FIG. 15 shows chip sequences for different DSSS values and polarities, according to an embodiment of the present disclosure. DSSS modulator 1300 may convert input bits to chips in accordance to one or more of the tables shown in FIG. 15. In some embodiments, DSSS modulator 1300 may implement two or more of the tables shown in FIG. 15 as a selectable set (e.g., based on a selected DSSS value and/or polarity).

[0201] As can be seen in FIGS. 14 and 15, a particular DSSS value and polarity may correspond to a different sequence of chips, e.g., depending on the particular implementation. For example, in some embodiments, for an input bit equal to 1, each of the corresponding pair(s) of chips should have the same chip value, and for an input bit equal to 0, each of the corresponding pair(s) of chips should have different chip values (although the order of the chip values may be implemented in different manners, e.g., as shown in FIGS. 14 and 15). Other implementations are also possible.

[0202] As shown in FIGS. 14 and 15, in some embodiments, each input bit is converted into a sequence of chips. In some embodiments, multiple input bits may be converted to a sequence of chips, where the input sequence has a length r (r being greater than 1), and the corresponding sequence of chips has a length s (s being greater than r). For example, some embodiments implement a two-to-eight DSSS, which converts two bits into a sequence of eight chips. In some such embodiments, each set of two input bits may have four possible values, where each of the four values is associated

with at least one unique sequence of eight chips. As other examples, some embodiments may implement a two-to-sixteen DSSS and/or a four-to-sixteen DSSS for communication devices implementing DSSS. Other implementations are also possible.

[0203] In some embodiments, an N-bit input sequence input to DSSS modulator 1300 is converted to a sequence of $N \times$ (DSSS value) binary valued chips.

[0204] FIG. 16 shows a block diagram of DSSS demodulator 1600, according to an embodiment of the present disclosure. Inverse DSSS blocks 508 and 604 may be implemented as DSSS demodulator 1600.

[0205] As shown in FIG. 16, DSSS demodulator 1600 may operate with symbols (e.g., as in inverse DSSS block 508) or with bits (e.g., as in inverse DSSS block 604).

[0206] In some embodiments, DSSS demodulator 1600 operates in an inverse manner as the DSSS modulator used for modulating packet 401 (e.g., with the same DSSS value, polarity, and chip conversion table(s)) so as to recover the original sequence of input bits. For example, in an embodiment using DSSS equal to 2 and an even polarity, an input sequence 0 1 1 0 may be converted by DSSS modulator 1300 into the sequence of chips: 0 0 0 1 0 1 0 0. DSSS block 508 implementing DSSS demodulator 1600 receiving such sequence of bits, and configured with DSSS equal to 2 and an even polarity, generates the despread bit sequence 0 1 1 0 (e.g., by comparing each pair of bits to determine if a bit is corrupted). As another example, DSSS block 604 implementing DSSS demodulator 1600 receiving sequence of symbols $S_0 S_0 S_0 S_1 S_0 S_1 S_0 S_0$, and similarly configured, generates the despread symbol sequence $S_0 S_1 S_1 S_0$, where S_0 and S_1 represent the (e.g., OFDM) modulated symbol corresponding to a bit of 0 and 1, respectively.

[0207] FIG. 17 shows a block diagram of BPSK encoder 1700, according to an embodiment of the present disclosure. Bits-to-symbols block 410 may be implemented as BPSK encoder 1700.

[0208] In some embodiments, BPSK encoder 1700 uses BPSK to map each input bit (e.g., each chip of input chip stream) into a symbol, e.g., according to the following BPSK encoding:

Input chip	BPSK symbol
0	$-1 + (0 \times j)$
1	$1 + (0 \times j)$

Other implementations are also possible.

[0209] In some embodiments, the symbol duration Sym-Dur of each symbol produced by BPSK encoder 1700 is fixed. In some embodiments, the symbol duration may be selectable from a set that includes: 120 μ s, 60 μ s, 30 μ s, and 15 μ s. Other symbol durations are also possible.

[0210] FIG. 18 shows a block diagram of BPSK decoder 1800, according to an embodiment of the present disclosure. Symbols-to-bits metrics block 512 and symbols-to-bits block 602 may be implemented as BPSK decoder 1800.

[0211] BPSK decoder 1800 may be implemented in an inverse manner as BPSK decoder 1700 so as to recover the bit that was used to generate the symbol by BPSK encoder 1700.

[0212] FIG. 19 shows a block diagram of single sub-carrier mapper 1900, according to an embodiment of the

present disclosure. Single sub-carrier mapping block **412** may be implemented as single sub-carrier mapper **1900**.

[0213] In some embodiments, single sub-carrier mapper **1900** maps each symbol of the input symbol stream to a single sub-carrier channel of the available set of (e.g., OFDM) channels. For example, when sub-channel select signal CH_{sel} indicates sub-channel **0**, the next symbol of the input symbol stream is mapped to the sub-channel **0**. More generally, if a set of available channels has L, when channel select signal CH_{sel} indicates sub-channel i ($0 < i < L$), the next symbol of the input symbol stream is mapped to sub-channel i.

[0214] In some embodiments, for each time slot, single sub-carrier mapper **1900** maps a single symbol to a single sub-carrier by modulating a sine or cosine wave or switching between sine or cosine waves. In some embodiments, the total communication bandwidth of RF core **300** may be divided into multiple (e.g., P) frequency bands/channels (also referred to as active tones), and RF core **300** transmits via antenna **308** on one of the frequency bands at a time. In some embodiments, single sub-carrier mapper **1900** may select the same sub-channel for each pair of symbols, such that differentially encoded symbols (e.g., DBPSK) share the same sub-carrier. In some embodiments, P may be greater than L. In some embodiments, P may be equal to L.

[0215] In some embodiments, the spacing between channels CH_{spc} is fixed. In some embodiments, the spacing between channels CH_{spc} is selectable from a set that includes: 200 kHz, 400 kHz, 800 kHz, and/or 1200 kHz. In some embodiments, different sub-channel spacings may be used.

[0216] In some embodiments, P may configurable and may depend based on a particular setting of RF core **300**. For example, in some embodiments, P is selectable from a set that includes: 12 channels, 26 channels, 52 channels, and/or 104 channels.

[0217] In some embodiments, the frequency band used for transmitting one or more symbols is predetermined (and may be fixed, e.g., CH_{sel} may be fixed, e.g., for an entire packet or groups of packets). In some embodiments, single sub-carrier mapper **1900** selects which frequency band to transmit one or more symbols based on a predetermined set of frequency bands, e.g., according to a (e.g., predetermined) hopping sequence. In some embodiments, single sub-carrier mapping block **310** selects which frequency band to send the symbols based on which portion of PPDU **400** is being transmitted.

[0218] FIG. 20 shows a block diagram of IFFT block **2000**, according to an embodiment of the present disclosure. Inverse transform block **414** may be implemented as IFFT block **2000**.

[0219] In some embodiments, IFFT block **2000** computes an IFFT on the symbols of input symbol stream to generate a sequence of (e.g., time-domain) samples corresponding to the input symbols. In some embodiments, the number of samples per symbol DFT_{sz} is fixed. In some embodiments, the number of samples per symbol DFT_{sz} is selectable from a set that includes: 16, 32, 64, and/or 128 samples per symbol. Other numbers of sample per symbol (e.g., powers of 2) may also be used.

[0220] FIG. 21 shows a block diagram of FFT block **2100**, according to an embodiment of the present disclosure. Fourier Transform block **504** may be implemented as FFT block **2100**.

[0221] In some embodiments, FFT block **2100** performs a Fourier transform on the input sample stream in an inverse manner as IFFT block **2000** (e.g., with the same DFT_{sz} setting) so as to reconstruct the transmitted symbols used by IFFT block **2000** to generate the time-domain signal with the generated sequence of samples.

[0222] FIG. 22 illustrates the operation of cyclic prefix with windowing function, according to an embodiment of the present disclosure. Cyclic prefix block **416** may implement a cyclic prefix with windowing function as shown in FIG. 22.

[0223] In some embodiments, cyclic prefix block **416** repeats all or a portion of each symbol before and/or after such symbol. For example, in some embodiments, each of the symbols is prepended and/or appended with a cyclic prefix (CP), e.g., as shown in FIG. 22.

[0224] In some embodiments, the duration of the CP may be % of the base symbol duration, but other durations, such as $\frac{1}{2}$ or different may also be used.

[0225] In some embodiments, the sum of the duration of the CP and the base symbol duration results into the overall symbol duration. For example, in some embodiments in which the duration of the CP is $\frac{1}{4}$ of the base symbol, the overall number of samples for each symbol may be given by:

$$SZ = DFT_{sz} + \frac{DFT_{sz}}{4} \quad (3)$$

where DFT_{sz} represents the number of samples of the base symbol (e.g., generated by IFFT **2000**). The overall symbol duration may be given by SZ times the base symbol duration. This is illustrated, e.g., in Table 1, according to an embodiment of the present disclosure.

TABLE 1

Cyclic prefix duration		
Base Symbol Duration	CP Duration	Symbol Duration
96 us	24 us	120 us
48 us	12 us	60 us
24 us	6 us	30 us
12 us	3 us	15 us

[0226] In some embodiments, to minimize the sidelobes of each of the subcarriers, a windowing filter may be applied during the first half of the Cyclic Prefix duration. For example, in some embodiments in which the symbol duration is $\frac{1}{4}$ of the base symbol, the windowing function may be applied to the first $DFT_{sz}/8$ samples of the CP).

[0227] In some embodiments, a windowing filter is not be applied during the cyclic prefix duration.

[0228] In some embodiments, for each current symbol k, the linear windowing function may combine samples of OFDM symbol k-1 together with samples of CP of symbol k. For example, in some embodiments in which the symbol duration is $\frac{1}{4}$ of the base symbol, the first samples of OFDM symbol k-1 together with first $DFT_{sz}/8$ samples of the Cyclic Prefix of symbol k. In some embodiments, this combination may be linear, so that the samples from OFDM symbol k-1 are multiplied with a linear function that starts at 1 at sample 0 and stops at 0 at sample $DFT_{sz}/8$ and samples from the Cyclic Prefix and multiplied with a linear

function that starts at 0 at sample 0 and stops at 1 at sample $DFT_{sz}/8$. The result of the two multiplications may be added to produce the new Cyclic Prefix samples. Portions of this process are illustrated in FIG. 22.

[0229] In some embodiments, the base symbol duration is predetermined. In some embodiments the base symbol duration may dynamically change, e.g., based on a data received (e.g., via a previous transmission between devices 202 and 252). For example, in some embodiments, the base symbol duration may be selected dynamically from a set of base symbol durations, such as shown in Table 1.

[0230] In some embodiments, RF core 300 supports multiple options for spreading rates and multiple symbol rates. In some embodiments, the selected option determines the overall signal bandwidth (BW). In some embodiments, the bandwidth of each of the sub-carriers is based on the selected symbol duration. In some embodiments, the spreading rate (e.g., the DSSS value) provides an additional coding mechanism to increase redundancy and improve link budget.

[0231] FIGS. 23A-23D show data rates for various possible settings of RF core 300, according to an embodiment of the present disclosure.

[0232] In some embodiments, the overall bandwidth of all frequency bands in which data can be encoded for transmission (e.g., under any of the settings shown in FIGS. 23A-23D) is 500 kHz (or greater), which may advantageously enable the use of asynchronous protocols in FCC regions. In some embodiments, the overall bandwidth may be lower than 500 kHz.

[0233] As shown in FIGS. 23A-23D, in some embodiments, RF core 300 may have multiple configurable parameters. For example, in an embodiment in which RF core 300 supports all of the parameter options shown in FIGS. 23A-23D, the symbol duration SymDur may be selectable from a set that includes 120 μ s, 60 μ s, 30 μ s, and 15 μ s. In some such embodiments, when symbol duration SymDur of 120 μ s is selected, RF core 300 may be configured as option 1, option 2, option 3, or option 4, e.g., as shown in FIG. 23A. In some such embodiments, if option 1 is selected, the nominal bandwidth for communication is 1.1 MHz, with a channel spacing (e.g., overall channel spacing, e.g., referring to the collection of all sub-carriers, where the sub-channel spacing may be given by $(5/4)*SymDur$) of 1.2 MHz, with a DFT size DFT_{sz} of 128 samples, with 104 sub-carrier channels (also referred to as active tones), and with a DSSS value that is selectable between 2, 4, and 6, selection of which affects the data rate (with higher DSSS values resulting in lower data rates).

[0234] As can be seen in FIGS. 23A-23D, for some symbol durations SymDur, some options may be unavailable. For example, as shown in FIG. 23D, for a symbol duration SymDur of 15 μ s, options 2, 3, and 4 may be unavailable.

[0235] In some embodiments, the general relationship between the data rate, as a function of the symbol duration SymDur and DSSS value may be given by:

$$\text{Data rate} = \frac{1}{2 \times \text{DSSS Value} \times \text{Sym Dur}} \quad (4)$$

[0236] In some embodiments, selecting an option and symbol duration may be negotiated (e.g., in advance) by upper protocol layers (e.g., above the PHY layer, such as by

the MAC layer, or above) of the devices (e.g., 202 and 252). In some embodiments, the spreading rate (the DSSS value) may be dynamically expressed per packet (e.g., including for specific portions within a packet), such as in a header of the packet.

[0237] In some embodiments, RF core may not use any pilot tones (e.g., may not use any channels for transmission of pilot signals, e.g., for synchronization).

[0238] In some embodiments, RF core 300 may support all of the settings shown in FIGS. 23A-23D. In some embodiments, RF core 300 may support other settings (not shown in FIGS. 23A-23D). In some embodiments, RF core 300 may some, but more than 1, of the settings shown in FIGS. 23A-23D. In some embodiments, RF core 300 may support only a single setting (e.g., any of the settings) of the settings shown in FIGS. 23A-23D.

[0239] As an example, in some embodiments, RF core 300 may support all of the option 2 settings listed in FIGS. 23A-23C (e.g., having DSSS values selectable between 2, 4, and 6; having symbol duration selectable between 120 μ s, 60 μ s, and 30 μ s; and having the DFT size DFT_{sz} selectable depending from a set that depends on the symbol duration SymDur, e.g., as shown in FIGS. 23A-23C).

[0240] In some embodiments, communication between device 202 and 252 use sub 1-GHz bands, such as bands between 470 MHz and 925 MHz. For example, some embodiments use one of the following bands for transmitting and/or receiving packets:

- [0241] 470-510 MHz;
- [0242] 779-787 MHz;
- [0243] 863-870 MHz;
- [0244] 865-867 MHz;
- [0245] 866-869 MHz;
- [0246] 870-876 MHz;
- [0247] 902-928 MHz;
- [0248] 902-928 (alternate) MHz;
- [0249] 902-907.5 MHz and 915-928 MHz;
- [0250] 915-928 MHz;
- [0251] 915-921 MHz;
- [0252] 915-918 MHz;
- [0253] 917-923.5 MHz;
- [0254] 919-923 MHz;
- [0255] 920-928 MHz;
- [0256] 920.5-924.5 MHz; and
- [0257] 920-925 MHz.

[0258] FIG. 24 shows a flow chart of embodiment method 2400, for selecting communication parameters, according to an embodiment of the present disclosure. In some embodiments, RF core 300, or an associated controller (e.g., 206, 256) may implement method 2400.

[0259] During step 2402, a first device (e.g., 202), negotiates with a second device (e.g., 252) to select a symbol duration to be used during a communication phase between the first and second devices. In some embodiments, the first device negotiates with the second device by communicating with the second device using a default setting (e.g., SymDur=60 μ s; option 2, according to FIG. 23B).

[0260] In some embodiments, both the first and second devices support various symbol durations (e.g., such as shown in FIG. 23A-23D). In some such embodiments, the symbol duration may be selected during step 2404 from a set of commonly supported symbol durations, e.g., based on the nature of the devices, environmental conditions, the nature of the particular application, etc. In some embodiments, if

the set of commonly supported symbol durations only has one symbol duration, such symbol duration is selected during step 2404.

[0261] During step 2406, the first and second devices negotiate to select an option associated with the selected symbol duration, where selection of an option may imply selection of a DFT size DFT_{sz} , number of active tones, etc., e.g., as shown in FIGS. 23A-23D. In some embodiments, both the first and second devices support various options (e.g., such as shown in FIG. 23A-23C). In some such embodiments, the option may be selected during step 2408 from a set of commonly supported options, e.g., based on the nature of the devices, environmental conditions, the nature of the particular application, etc. In some embodiments, if the set of commonly supported options only has one symbol duration (e.g., as shown in FIG. 23D, or if a device only supports option 2 of FIG. 23B), such option is selected during step 2408.

[0262] During step 2410, the first and second devices communicate using the parameters selected during steps 2404 and 2408.

[0263] FIG. 25 illustrates a packet structure of PPDU 2500, according to an embodiment of the present invention. Packet 401 may be implemented as PPDU 2500.

[0264] PPDU 2500 includes short training field (STF) 2502, long training field (LTF) 2504, physical header (PHR) field 2506, and PHY payload field 2508. PHR field 2506 includes rate field 2520, frame length field 2522, header check sequence (HCS) field 2526, and tail field 2526. PHY payload field 2508 includes PHY service data unit (PSDU) field 2540, PPDU tail field 2542, and pad field 2544.

[0265] In some embodiments, STF field 2502 may be used by RF core 300 (e.g., demodulator 304, e.g., processing pipelines 500 or 600) or an associated controller (e.g., 206, 256) for packet detection and/or for (e.g., coarse) frequency offset determination. In some embodiments, a portion of STF field 2502 may be configurable. For example, in some embodiments, that last bit of STF field 2502 may be configurable. In some such embodiment, the portion of the STF field 2502 used for packet detection and/or frequency offset determination may be a fixed (non-configurable) portion of STF field 2502, while the configurable portion of STF field 2502 may not be used for packet detection and/or for frequency offset determination.

[0266] In some embodiments, STF field 2502 includes 160 symbols. In some embodiments, STF field 2502 may include fewer than 160 symbols (e.g., 120, 64, or lower), or more than 160 symbols (e.g., 200, 320, or more).

[0267] In some embodiments, the symbols of STF field 2502 are transmitted in an uninterrupted manner via antenna 308 in a single sub-carrier.

[0268] In some embodiments, the content of STF field 2502 is processed by some, but not all, of the blocks of processing pipeline 400. For example, in some embodiments, blocks 402, 404, and 406 do not process the content of STF field 2502. In some such embodiments, the content of STF field 2502 is processed by DSSS block 408 (e.g., with a predetermined DSSS value, such as 2), bits-to-symbols block 410 (e.g., using BPSK), single sub-carrier mapping block 412 (e.g., with CH_{sel} fixed), and inverse transform block 414. In some such embodiments, cyclic prefix block 416, and filtering block 418 also process the content of STF field 2502.

[0269] In some embodiments, the content of STF field 2502 is fixed. For example, FIG. 26, shows STF bit sequence 2600, according to an embodiment of the present disclosure. Other STF bit sequences may also be used.

[0270] As shown in FIG. 26, in some embodiments, the STF bit sequence may have 80 bits. Such STF sequence may be doubled when using a DSSS value of 2 so that the sequence transmitted by antenna 308 may include a multiple of the number of bits of the original STF sequence (e.g., 160 chips, when using DSSS equal to 2).

[0271] In some embodiments, STF field 2502 includes one or more configurable bits.

[0272] In some embodiments, some or all (e.g., BPSK) symbols of STF field 2502 may be used to generate a series of vectors

$$X_k = \{x_k(n)\}, -\frac{DFT_{sz}}{2} \leq n \leq \frac{DFT_{sz}}{2} - 1,$$

with k between 0 and e.g., the total number of symbols of STF field 2502-1 (e.g., from 0 to 159 when STF field 2502 has 160 symbols), where DFT_{sz} is defined based on the option selected (e.g., using method 2400), and where $x_k(STF_{subcarrier})=b(k)$ and $x_k(n \neq STF_{subcarrier})=0$, where STF_{subcarrier} is the (e.g., single) CH_{sel} used by single sub-carrier mapper 1900 when processing (e.g., all content of) STF field 2502.

[0273] In some embodiments, LTF field 2504 may be used by RF core 300 (e.g., demodulator 304, e.g., processing pipelines 500 or 600) or an associated controller (e.g., 206, 256) for finer frequency offset detection, for time synchronization, and/or for channel equalization.

[0274] In some embodiments, LTF field 2504 includes fewer number of symbols than STF field 2502. In some embodiments, LTF field 2504 includes 26 symbols. Some embodiments may include more than 26 symbols (e.g., 32, 40, or more) or less than 26 symbols (e.g., 20, 16, or less).

[0275] In some embodiments, the symbols of LTF field 2504 are transmitted sequentially via antenna 308 in multiple sub-carriers (only using a single sub-carrier at a time) in accordance with a hopping sequence.

[0276] In some embodiments, the content of LTF field 2504 is processed by some, but not all, of the blocks of processing pipeline 400. For example, in some embodiments, blocks 402, 404, and 406 do not process the content of LTF field 2504. In some such embodiments, the content of LTF field 2504 is processed by DSSS block 408 (e.g., with a predetermined DSSS value, such as 2), bits-to-symbols block 410 (e.g., using BPSK), single sub-carrier mapping block 412 (e.g., with CH_{sel} fixed or changing in accordance with a hopping sequence), and inverse transform block 414. In some such embodiments, cyclic prefix block 416, and filtering block 418 also process the content of LTF field 2504.

[0277] In some embodiments, the content of LTF field 2504 is fixed. For example, FIG. 27, shows LTF bit sequence 2700, according to an embodiment of the present disclosure. Other LTF bit sequences may also be used.

[0278] In some embodiments, LTF field 2504 includes one or more configurable bits.

[0279] In some embodiments, PHR field 2506 may be used by RF core 300 (e.g., demodulator 304, e.g., processing

pipelines **500** or **600**) or an associated controller (e.g., **206**, **256**) for determining the format, modulation, and/or content of PHY payload field **2508**.

[0280] In some embodiments, the symbols of PHR field **2506** are transmitted sequentially via antenna **308** in multiple sub-carriers (only using a single sub-carrier at a time) in accordance with a hopping sequence.

[0281] In some embodiments, the content of PHR field **2506** is processed by some, but not all, of the blocks of processing pipeline **400**. For example, in some embodiments, blocks **402** does not process the content of LTF field **2504**. In some such embodiments, the content of PHR field **2506** is processed by FEC encoder block **404**, interleaver block **406**, DSSS block **408** (e.g., with a predetermined DSSS value, such as **6**), bits-to-symbols block **410** (e.g., using BPSK), single sub-carrier mapping block **412** (e.g., with CH_{sel} fixed or changing in accordance with a hopping sequence), and inverse transform block **414**. In some such embodiments, cyclic prefix block **416**, and filtering block **418** also process the content of PHR field **2506**.

[0282] In some embodiments, the input of scrambler block **402** include (e.g., some or all of) the data octets of PHR field **2506**.

[0283] In some embodiments, PHR field **2506** includes 24 bits. In some embodiments, PHR field includes a different number of bits, such as more than 24 bits, or less than 24 bits.

[0284] In some embodiments, some of the bits of PHR field **2506** are fixed and some are configurable. In some embodiments, all bits of PHR field **2506** are configurable.

[0285] In some embodiments, interleaver block **406** performs interleaving to (e.g., some or all bits of) PHR field **2506**.

[0286] In some embodiments, PHY payload field **2508** includes data bits.

[0287] In some embodiments, PHY payload field **2508** has a variable length (e.g., as indicated by frame length field **2522**), so each packet may have a different length of PHY payload field. In some embodiments, PHY payload field **2508** has a fixed (e.g., predetermined) length.

[0288] In some embodiments, some or all of the bits of PHY payload field **2508** are transmitted at a data rate specified by PHR **2506**. In some embodiments, a data rate used for transmission of PHY payload field **2508** is the same as the data rate used for transmission of PHR field **406**. In some embodiments, the data rate used for transmission of PHY payload field **2508** is different from the data rate used for transmission of PHR field **2506**.

[0289] In some embodiments, convolutional encoder **1000** is initialized to the all zeros state before encoding the bits associated with PHR field **2506** and then reset to the all zeros states before encoding bits of PHY payload field **2508**.

[0290] In some embodiments, rate field **2520** indicates the DSSS value used for encoding PHY payload field **2508** (and, thus, may impact the data rate in which bits of PHY payload field **2508** are transmitted). In some embodiments, the DSSS value indicated by rate field **2520** is a DSSS value from a predetermined set of DSSS values. In some embodiments, the predetermined set of DSSS values include DSSS values of 2, 4, and/or 6. In some embodiments, the predetermined set of DSSS value may include other DSSS values (e.g., 0 (DSSS not used), 8, 10, 12, etc.).

[0291] In some embodiments, rate field **2520** is a 2-bit field. In some embodiments, rate field may include more

than 2 bits (e.g., 3, 4, or more), or 1 bit. In some embodiments, the content of rate field **2520** may be fixed. In some embodiments, some or all bits of frame length field **2520** are configurable. In some embodiments, rate field **2520** is not implemented (omitted). In some such embodiments, the DSSS value used for PHY payload **2508** may be fixed.

[0292] In some embodiments, frame length field **2522** is indicative of the total length of PPDU **2500**, and/or of a portion of PPDU **400**, such as a length of PHY payload field **2508** and/or the length of portions of PHY payload field **2508**, such as PSDU field **2540**, PPDU tail field **2542**, and/or pad field **2544**.

[0293] In some embodiments, the content of frame length field **2522** is an unsigned integer that indicates a (e.g., total) number of octets (bytes) contained in PSDU field **2540** (e.g., prior to encoding by FEC encoder block **404**).

[0294] In some embodiments, frame length field **2522** is an 8 bits field. In some embodiments, length field **2522** may be a field having more than 8 bits (e.g., 9, 10, 20, or more), or less than 8 bits (e.g., 6, 4, 3, 2, or 1). In some embodiments, the content of frame length field **2522** may be fixed. In some embodiments, some or all bits of frame length field **2522** are configurable. In some embodiments, frame length field **2522** is not implemented (omitted). In some such embodiments, the length of PHY payload field **2508** (and possible of PPDU **2500**) may be fixed.

[0295] In some embodiments, frame length field **2522** is transmitted with a most significant bit (MSB) first. In some embodiments, frame length field **2522** is transmitted with a least significant bit (LSB) first.

[0296] In some embodiments, HCS field **2524** is a cyclic redundancy check (CRC) field. In some embodiments, such CRC is for checking the transmission of PHR field **2506** (e.g., for detecting/correcting errors of PHR field **406**). In some embodiments, such CRC is computed based on content of rate field **2520** and frame length field **2522**. In some embodiments, HCS field **2524** is computed using the first 10 bits of PHR field **2506**.

[0297] In some embodiments, RF core **300** (e.g., demodulator **304**, e.g., processing pipelines **500** or **600**) or an associated controller (e.g., **206**, **256**) may compare the value received in HCS field **2524** with a CRC value computed based on content of rate field **2520** and frame length field **2522** to determine whether PHR field **2506** is corrupted.

[0298] In some embodiments, HCS field **2524** may be computed using the following polynomial:

$$G_8(x) = x^8 + x^2 + x + 1. \quad (5)$$

[0299] In some embodiments, HCS field **2524** is the one's complement of the modulo-2 sum of the two remainders in a) and b):

[0300] a) the remainder resulting from $[x^k(x^7+x^6+\dots+1)]$ divided (modulo 2) by $G_8(x)$, where the value k is the number of bits in the calculated field;

[0301] b) the remainder resulting from the calculation field content, treated as a polynomial, multiplied by x^8 and then divided (modulo 2) by $G_8(x)$.

[0302] In some embodiments, HCS field **2524** is an 8 or 10 bit field. In some embodiments, HCS field **2524** may be a field having more than 8 bits (e.g., 9, 12, 16, or more), or less

than 8 bits (e.g., 7, 6, or less). In some embodiments, HCS field **2524** is not implemented (omitted).

[0303] In some embodiments, tail field **2526** is intended to facilitate decoding, e.g., by FEC decoder **514**, to reduce error probability, e.g., of a convolutional encoder (e.g., **1000**). For example, in some embodiments, tail field **2526** includes all zeros to facilitate Viterbi decoded flushing.

[0304] In some embodiments, tail field **2526** is a 6 bit field. In some embodiments, tail field **2526** may include more than 6 bits (e.g., 7, 9, 10, or more), or less than 6 bits (e.g., 5, 4, or less). In some embodiments, tail field **2526** may be omitted.

[0305] In some embodiments, PHR field **2506** includes a scrambler field (not shown) to specify a scrambler seed used for scrambling (e.g., by blocks **402**, **516**) content of PHY payload field **2508**. In some embodiments, the scrambler field **2506** may be omitted. In some such embodiments, scrambling of content of PHY payload field **2508** may be based on a predetermined scrambling seed (e.g., all ones) or may be omitted.

[0306] In some embodiments, PSDU field **2540** includes data bits. In some embodiments, such data bits of PSDU field **2540** are data bits received from a MAC layer of the device (e.g., **202**, **252**) performing the transmitting (e.g., using processing pipeline **400**).

[0307] In some embodiments, the input of scrambler block **402** include (e.g., some or all of) the data octets of PSDU field **2540**.

[0308] In some embodiments, PSDU field **2540** has a variable length (e.g., as indicated by frame length field **2522**). In some embodiments, the length of PSDU field **2540** is based on the number of data bits received from the MAC layer.

[0309] In some embodiments, content of PSDU field **2540** is transmitted at a data rate specified by PHR **2506** (e.g., data rate **2520**). In some embodiments, a data rate used for transmission of PSDU field **2540** is the same as the data rate used for transmission of the rest of PHR payload field **2508**, and/or the PHR field **2506**. In some embodiments, the data rate used for transmission of PSDU field **2540** is different from the data rate used for transmission of the rest of PHR payload field **2508**, and/or the PHR field **2506**.

[0310] In some embodiments, PPDU tail field **2542** is intended to facilitate decoding, e.g., by FEC decoder **514**, to reduce error probability, e.g., of a convolutional encoder (e.g., **1000**). For example, in some embodiments, PPDU tail field **2542** includes all zeros to facilitate Viterbi decoded flushing.

[0311] In some embodiments, PPDU tail field **2542** is a 6 bit field. In some embodiments, tail field **2542** may include more than 6 bits (e.g., 7, 8, 10, or more), or less than 6 bits (e.g., 5, 4, or less). In some embodiments, PPDU tail field **2542** may be omitted.

[0312] In some embodiments, a processing pipeline (e.g., **500** or **600**) of a demodulator (e.g., **304**) places (e.g., some or all) fields of reconstructed packet **401** (e.g., fields **2502**, **2504**, **2506**, and/or **2508**) into a receiver (RX) first-in-first-out (FIFO) buffer for further processing/use (e.g., by demodulator **304** and/or an associated controller (e.g., **206**, **256**)).

[0313] FIG. 28 illustrates transmission of packet **2800** encoded across multiple sub-carrier frequencies, according to an embodiment of the present disclosure. In some

embodiments, packet **401** may be transmitted as packet **2800**, e.g., by RF core **300**, e.g., after processing by processing pipeline **300**.

[0314] In the embodiment of FIG. 28, 16 sub-carrier frequencies are used. Some embodiments may use less than 16 sub-carrier frequencies for packet transmission (e.g., 12) or more than 16 sub-carrier frequencies, e.g., 26, 52, or 104. In some embodiments, the number sub-carrier frequencies used is determined by the option used for communication, e.g., as determined by method **2400**.

[0315] In some embodiments, packet **2800** has the packet structure of PPDU **2500**.

[0316] As shown in FIG. 28, packet **2800** may be transmitted by first transmitting synchronization sequence **2802** (e.g., corresponding to STF field **2502**), immediately followed by the rest of the fields of packet **2800** (e.g., fields **2504**, **2506**, **2508**), e.g., in accordance with hopping sequence **2804**.

[0317] As shown in FIG. 28, transmission of packet **2800** may take t_{packet} time, which includes t_{sync} time (time for transmission synchronization sequence **2802**) and t_{hop} time (time for transmitting data according to hopping sequence **2804**).

[0318] In FIG. 28, pairs of symbols **2810-2850** are illustrated as two horizontally adjacent boxes. The horizontal axis in FIG. 28 represents the time slots for transmitting data. The vertical axis in FIG. 28 represents the sixteen frequency bands in which data can be encoded for transmission. Although FIG. 3E illustrates transmission using 16 frequency bands, some embodiments may use more than 16 frequency bands (e.g., 20, 26, 32, 37, 40, 52, 104, or more, or less than 16 frequency bands, e.g., 14, 12, 9, 8, 4, or less). In some embodiments, the set of frequency bands available for transmission may be dynamically selected.

[0319] In some embodiments, the frequency bands used for hopping sequence **2804** is a subset of frequency bands (e.g., with fewer frequency bands) of all frequency bands available for communication (e.g., the number of active tones, e.g., according to the communication option selected, e.g., from FIGS. 23A-23D). For example, in some embodiments, there may be a total of 16 frequency bands available for communication (e.g., 16 active tones) but the hopping sequence uses only 10 of the 16 frequency bands. Other implementations are also possible.

[0320] As shown in FIG. 28, some embodiments use a single sub-carrier signal for each time slot. This technique may advantageously result in lower PAPR and produce an RF signal that may be easier for a demodulator (e.g., **304**) to demodulate.

[0321] As shown in FIG. 28, some embodiments select keep the same sub-carrier frequency for the entirety of synchronization sequence **2802**, and select a new sub-carrier frequency after every two time slots (every two symbols—every pair of symbols) so that the symbols in each pair share the same frequency band (which may be advantageous for differential modulations, such as differential BPSK (DBPSK)). Some embodiments may use FDM to divide up the total bandwidth into channels that can be selected for each pair of time slots. Additionally or alternatively, some embodiments may implement other forms of division multiplexing, such as time division multiplexing or code division multiplexing, for modulating RF signals.

[0322] In the embodiment of FIG. 28, pair of symbols **2810** is assigned to the ninth frequency band, pair of

symbols **2812** to the sixth frequency band, and so on. Each pair of symbols is assigned to a frequency band (e.g., fixed or according to a hopping sequence).

[0323] In some embodiments (such as shown in FIG. 28), the first frequency channel of the hopping sequence **2804** is the same as STFsubcarrier (the frequency channel used for transmission of the synchronization sequence **2802**). In some embodiments, the first frequency channel of the hopping sequence **2804** may be different than STFsubcarrier.

[0324] In some embodiments, more than 1 pair of symbols (e.g., 2, 3, 4, or more) may be transmitted before hopping to the next channel, which may advantageously result in less frequent channel switching than in the embodiment of FIG. 28. In some embodiments, hopping sequence **2804** is random, pseudo-random, quasi-random, or deterministic.

[0325] In some embodiments, hopping sequence **2804** may switch frequencies for each hop (e.g., after each pair of symbols). However, in some embodiments, the hopping sequence **2804** may occasionally select the same frequency band for two consecutive transmissions (e.g., as illustrated by pairs of symbols **2820** and **2822**). For example, in some embodiments using random, pseudo-random, or quasi-random hopping sequences, the same sub-channel may be selected for consecutive transmissions. In some embodiments, hopping sequence **2804** is designed to prevent consecutive transmissions using the same frequency band. In some embodiments, hopping sequence **2804** is designed to hop through all channels before revising a particular sub-channel (and such sequence may be repeated until all symbols of packet **2800** are transmitted).

[0326] In some embodiments, a demodulator (e.g., **304**) determines the bits represented by each pair of symbols (e.g., inverse DSSS blocks **508**, **604**) by comparing the two symbol values in each pair. For example, in some embodiments using a DSSS value of 2, a logical value of one is represented by two identical symbols values, and a logical value of zero is represented by two non-identical symbol values. Thus, in some embodiments, the demodulator (e.g., inverse DSSS blocks **508**, **604**) can compare the second symbol value in the pair to the previous symbol value to determine the value of the bit represented by that pair of symbols. In some such embodiments, the demodulator can advantageously demodulate the RF signal using local information without any additional information, which may advantageously reduce the interference caused by frequency drift over time and multi-path environments.

[0327] In some embodiments, such as embodiments using BPSK, reusing the same frequency band for each symbol of a pair of symbols advantageously allows for differential encoding (e.g., DBPSK), which may advantageously allow for determining the phase of the symbol without precise synchronization, since the phase is relative and the frequency band is the same for both symbols of a pair of symbols.

[0328] In some embodiments, each symbol of packet **2800** transmitted corresponds to a chip. Thus, in some embodiments, each pair of symbols corresponds to a pair of chips. Thus, in some embodiments (e.g., using DSSS=2), each pair of symbols (e.g., **602**, **604**, etc.) corresponds to a bit of PPDU **2500**.

[0329] In some embodiments using DSSS values higher than 2 (e.g., 4 or higher), chips representing each bit are sent over multiple frequencies (e.g., according to a hopping sequence **2804**). Thus, some such embodiments may be

advantageously more robust and resilient over horizontal and vertical interference, as bits of PPDU **2800** are spread over time (e.g., multiple pairs of chips), and multiple frequency bands (e.g., due to hopping sequence **2804**). For example, assuming the embodiment of FIG. 28 implements a DSSS value of 4, pairs of symbols **2810** and **2812** represent a single bit of data (e.g., of PPDU **2800**). Such bit of data is sent over 4 time slots (4 symbols) and over multiple frequency bands (channels **9** and **6**).

[0330] In some embodiments, the higher the DSSS value, the more robust the communication is. In some embodiments, the lower the DSSS value, the higher the data throughput.

[0331] In some embodiments, the hopping sequence **2804** may be selected based on synchronization sequence **2802**. For example, in some embodiments, the seed for determining the hopping sequence **2804** may be based on which sub-channel is used for transmitting synchronization sequence **2802** (STFsubcarrier) and/or one or more bits of synchronization sequence **2802** (e.g., the last 2 bits of STF field **2502**).

[0332] In some embodiments, synchronization sequence **2802** is used for packet detection (e.g., detection of PPDU **2800**, such as detection of the start of LTF field **2504** for the beginning of hopping sequence **2804**), frequency offset determination, and/or for timing synchronization. In some embodiments, synchronization sequence **2802** is the only synchronization sequence sent for a packet. In some embodiments, a subsequent synchronization sequence (e.g., LTF field **2504**) is transmitted after synchronization sequence **2802** to allow for finer synchronization, timing synchronization, and/or channel equalization. For example, in some embodiments, such subsequent synchronization sequence may be sent following the hopping sequence (e.g., packets **2810**, **2812**, etc., may be part of LTF field **2504**), with the actual header (e.g., **2506**) and payload (e.g., **2508**) being transmitted after such subsequent synchronization sequence.

[0333] In some embodiments, synchronization sequence **2802** may include a preamble field, and a synchronization sequence field.

[0334] As shown in FIG. 28, in some embodiments, synchronization sequence **2802** is transmitted in a single frequency band. In some embodiments that include a subsequent synchronization sequence, the subsequent synchronization sequence is transmitted according to a hopping sequence **2804**. For example, in some embodiments, hopping sequence **2804** includes the subsequent synchronization sequence (LTF field **2504**). In some such embodiments, synchronization sequence **2802** (STF field **2502**) may be used for packet detection and coarse frequency offset determination, and the subsequent synchronization sequence (LTF field **2504**) may be used for finer offset determination, timing synchronization, and channel equalization.

[0335] In some embodiments, synchronization sequence **2502** includes a plurality of symbols, such as 10, 20, 32, 64, 80, 96, 160, or more.

[0336] In some embodiments, synchronization sequence **2802** has a time duration t_{sync} that is longer than a maximum sleep time duration (time spent in low power mode, e.g., with paths **310** and/or **320** disabled) of a receiver device (e.g., **252**). In some embodiments, the duration of synchronization sequence may change dynamically, e.g., by chang-

ing the duration of each symbol, and/or by changing the number of symbols transmitted as part of synchronization sequence **2802**.

[0337] In some embodiments, the receiver and/or transmitter device (e.g., **252**, **202**) periodically enters sleep mode, which may advantageously allow for lower power consumption. In some such embodiments, transmission/reception of packets (e.g., **401**) may occur when the devices are in active node, e.g., with paths **310** and **320** enabled.

[0338] In some embodiments, the duration of symbols for synchronization sequence **2802** (and/or of hopping sequence **2804**) is fixed. In some embodiments, the duration of symbols for synchronization sequence **2802** (and/or of hopping sequence **2804**) may change, e.g., periodically and/or in response to a trigger, such as an input from an upper layer (e.g., link layer, MAC layer, and/or transport layer).

[0339] In some embodiments, the subcarrier channel STF-subcarrier selected by single sub-carrier mapping block **412** for synchronization sequence **2802** may be selected by a link layer/MAC layer/network layer and may be fixed for multiple (or all) packets. In some embodiments, the subcarrier channel STFsubcarrier may change after each packet or after a plurality of packet (e.g., in response to a trigger, such as a trigger from a link layer/MAC layer/network layer or after a predetermined number of packets have been sent).

[0340] In some embodiments (such as illustrated in FIG. **28**), data transmission according to hopping sequence **2804** begins immediately after synchronization sequence **2802**. In some embodiments, there may be a delay (e.g., 1 or 2 symbols in duration or more) between synchronization sequence **2802** and the first transmission (e.g., **2610**) of the hopping sequence **2804**.

[0341] In some embodiments, a demodulator (e.g., **304**) may detect only a portion of synchronization sequence **2802** (e.g., the last portion) and may still be able to detect the packet (**2800**), perform synchronization, and receive data (e.g., fields **2504**, **2506**, and **2508**) during hopping sequence **2804**.

[0342] In some embodiments, the hopping sequence **2804** is selected based on which sub-channel the synchronization sequence **2802** is transmitted. For example, in the embodiment of FIG. **28**, a seed for selecting hopping sequence may be based on sub-channel **9**, which is the sub-channel in which synchronization sequence **2802** is transmitted. Thus, in the embodiment of FIG. **28**, a demodulator (e.g., **304**) may derive the hopping sequence used for data transmission by using (e.g., solely, or in combination with other parameters) sub-channel **9** as a seed for selecting the hopping sequence.

[0343] In some embodiments, the hopping sequence **2804** is selected based on one or more (e.g., of the last) symbols of the synchronization sequence **2802**. For example, in some embodiments, synchronization sequence **2802** may include one or more symbols, the state of which is indicative of a value used for determining a seed value for determining hopping sequence **2804**. Thus, in some embodiments, a demodulator (e.g., **304**) may derive the hopping sequence used for data transmission by using (e.g., solely, or in combination with other parameters) from the state of one or more symbols of synchronization sequence **2802**.

[0344] In some embodiments, the hopping sequence **2804** may be determined based on the state of one or more

symbols of synchronization sequence **2802** in combination with the sub-channel used for transmitting synchronization sequence **2802**.

[0345] In some embodiments, the hopping sequence **2804** is not based on content of synchronization sequence **2802** or the sub-channel in which synchronization sequence **2802** is transmitted. For example, in some embodiments, the hopping sequence **2804** is based on a seed transmitted out-of-band, a predefined seed, a current time, etc.

[0346] In some embodiments, the seed value for determining hopping sequence **2804** is partially determined by the frequency band in which synchronization sequence **2802** is transmitted in. In some embodiments, the seed value is fully determined by the frequency band in which synchronization sequence **2802** is transmitted in. In some embodiments, the seed value is not determined by the frequency band in which synchronization sequence **2802** is transmitted in.

[0347] In some embodiments, hopping sequence **2804** is selected based on a seed value in combination with one or more coefficients. In some embodiments, the seed value together with the one or more coefficients fully determine hopping sequence **2804**.

[0348] In some embodiments, the seed value and the one or more coefficients are predetermined, known to both devices (e.g., **202** and **256**) and fixed. In some embodiments, the seed value and/or one or more (or all) of the one or more coefficients may dynamically change (e.g., periodically, in response to an input received by RF core **300** and/or associated controller, after a predetermined number of transmission/packets, and/or other factors).

[0349] In some embodiments, one or more (or all) of the seed value and the one or more coefficients are received by the PHY layer (e.g., implemented by RF core **300**) from an upper layer (e.g., link layer, medium access control (MAC) layer, and/or network layer), which may be implemented, e.g., by an associated controller (e.g., **206**) or by another controller of the device (e.g., **202**).

[0350] In some embodiments, the one or more coefficients for determining the hopping sequence are determined using linear congruential generator (LCG) subcarrier mapping.

[0351] In some embodiments, sub-carrier mapping block **412** (e.g., single sub-carrier mapper **1900**) receives as input the one or more coefficients from a link layer/MAC layer/network layer. In some embodiments, the one or more coefficients are fixed and do not change. In some embodiments, one or more (or all) of the one or more coefficients change after each packet, after a predetermined number of packets transmitted, and/or in response to a trigger (e.g., received from a link layer/MAC layer/network layer).

[0352] Some embodiments include a seed value for determining hopping sequence **2804** based on STFsubcarrier and two coefficients, namely LCGa and LCGc. For example, if $S=\{s(k)\}$, $0 \leq k \leq M-1$, where $s(k) \in (0, 255)$ is a vector that contains all the seed values for corresponding hopping sequences, a second vector $J=\{j(k)\}$, $0 \leq k \leq M-1$, where

$$j(k) \in \left(-\frac{N_a}{2}, \frac{N_a}{2}\right),$$

N_a is the number of active tones (e.g., channels used for transmission) that contains the channel index to be used in that specific symbol, and the starting seed for generating the

hopping sequence (e.g., **2804**) be $s(0)=s(1)=DFT_{sz}+STF_{subcarrier}$, the starting seed for the algorithm, with $j(0)=j(1)=STF_{subcarrier}$.

[0353] In some embodiments, for all values of k above or equal to 2, a temporary (e.g., 8-bit) value s_{tmp} is initialized to be the seed value from the previous iteration $s(k-1)$. Then, a linear congruential random generator may be used to update the temporary seed value by

$$s_{tmp} = ((s_{tmp} \times LCG_a) + LCG_c) \bmod 256. \quad (6)$$

[0354] The P LSBs of s_{tmp} may then be extracted, where P may be given by:

$$P = 1 + \log_2 DFT_{sz}, \quad (7)$$

where DFT_{sz} is the number of samples of a symbol, to create a second temporary seed value by

$$r_{tmp} = s_{tmp} \bmod P \quad (8)$$

r_{tmp} is then compared against three conditions: r_{tmp} greater than or equal to

$$DFT_{sz} - \frac{N_a}{2};$$

r_{tmp} less than or equal to

$$DFT_{sz} + \frac{N_a}{2};$$

and r_{tmp} not equal to DFT_{sz} , where N_a is the number of data tones in a symbol. In some embodiments, when all 3 conditions are true, then $j(k)=r_{tmp}-DFT_{sz}$ and $s(k)=s_{tmp}$, otherwise, a new iteration is performed until the three conditions are satisfied.

[0355] In some embodiments, for each odd value of k , $s(k)=s(k-1)$ and $j(k)=j(k-1)$.

[0356] In some embodiments, for each (e.g., BPSK) symbol transmitted as part of hopping sequence **2804** (e.g., fields **2504**, **2506**, and **2508**), $B=\{b(k)\}$, $0 \leq k \leq M-1$, a vector

$$X_k = \{x_k(n)\}, -\frac{DFT_{sz}}{2} \leq n \leq \frac{DFT_{sz}}{2} - 1$$

is created, where $x_k(j(k))=b(k)$ and $x_k(n \neq j(k))=0$.

[0357] As an example, for 26 subcarriers (e.g., option 3 when SymDur=120 μ s (FIG. 23A); option 2 when SymDur=60 μ s (FIG. 23B); or option 1 SymDur=30 μ s (FIG. 23C), when STFsubcarrier=2, LCGa=17, LCGc=83, when DFT_{sz} is 32 (e.g., since 32 is the minimum number that is a power of 2 that fits the number of subcarriers (26)), the sequence generation outlined above provides the following Qa numbers:

[0358] 18 5 20 29 19 22 9 24 4 23 26 13 3 6 28 8 27 17
7 10 12 21 11 14 25 15.

[0359] As shown by numbers Qa, in some embodiments, the hopping sequence generated using sequence generation outlined above results in no numbers being repeated. The sequence Qa may be repeated as more symbols are transmitted (e.g., after 26 pairs of symbols have been transmitted, the next pair of symbols may use sub-channel **18** again).

[0360] As shown by numbers Qa, in some embodiments, the first number corresponds to STFsubcarrier=2+ $DFT_{sz}/2$.

[0361] As another example, for 26 subcarriers, when STFsubcarrier=2, LCGa=17, LCGc=83, when DFT_{sz} is 32, the sequence generation outlined above provides the following Qb numbers:

[0362] 19 22 9 24 4 23 26 13 3 6 28 8 27 17 7 10 12 21
11 14 25 15 18 5 20 29.

[0363] As can be seen by comparing sequences Qa and Qb, sequence Qb is a rotated version of sequence Qa, where the last 4 numbers of sequence Qb are the first 4 numbers of sequence Qa.

[0364] In some embodiments, the values of LCGa and LCGc come from respective predetermined sets of possible values. For example, FIGS. 29A and 29B show sets of possible values of LCGa and LCGc, respectively, according to an embodiment of the present disclosure.

[0365] In some embodiments, synchronization sequence **2802** is transmitted periodically.

[0366] In some embodiments, synchronization sequence **2802** is transmitted per packet **2800**. For example, in some embodiments, the first field to be transmitted for each packet **2800** is the STF field **2502**.

[0367] In some embodiments, the modulation scheme used for transmitting symbols of synchronization sequence **2802** is the same as the modulation scheme used for transmitting symbols hopping sequence **2804**. For example, all symbols of synchronization sequence **2802** and hopping sequence **2804** may be transmitted using BPSK (e.g., DBPSK). In some embodiments, the modulation scheme used for transmitting symbols of synchronization sequence **2802** may be different from the modulation scheme used for transmitting symbols of hopping sequence **2804**. For example, in some embodiments, symbols hopping sequence **2804** may be transmitted using BPSK (e.g., DBPSK) while symbols of synchronization sequence **2802** may be transmitted PSK, GFSK, BPSK, FSK, ASK, CSS, QAM, APSK, CPS, MSK or OOK.

[0368] In some embodiments, regardless of the modulation scheme used for symbol transmission, only a single sub-channel of the available sub-channels is used at a time for transmitting symbols, as illustrated in FIG. 28.

[0369] FIG. 30 shows transmission **3000** of a plurality of packets **401**, according to an embodiment of the present disclosure. Processing pipeline **300** may process packet each of the plurality of packets **401** of transmission **3000**.

[0370] As shown in FIG. 30, in some embodiments, each of the transmitted packets **401** has the same (e.g., fixed) synchronization sequence **2802a** (e.g., STF sequence shown in FIG. 2600) transmitted in the same sub-channel (e.g., sub-channel **9**). As also shown in FIG. 30, in some embodiments, the hopping sequence **2804** for each of the **401** packets is the same (although the transmitted data may be different).

[0371] FIG. 31 shows transmission **3100** of a plurality of packets **401**, according to an embodiment of the present

disclosure. Processing pipeline 300 may process packet each of the plurality of packets 401 of transmission 3100.

[0372] As shown in FIG. 31, in some embodiments, each of the transmitted packets 401 has the same (e.g., fixed) synchronization sequence 2802a (e.g., STF sequence shown in FIG. 2600) transmitted in different sub-channel for each packet. As also shown in FIG. 31, in some embodiments, the hopping sequence 2804 for each of the 401 packets may be different. In some embodiments, each of hopping sequences 2804b, 2804c, and 2804d is a rotated version of hopping sequence 2804a. In some embodiments, each of hopping sequences 2804a, 2804b, 2804c, and 2804d are different and are not rotated versions of each other.

[0373] FIG. 32 shows transmission 3200 of a plurality of packets 401, according to an embodiment of the present disclosure. Processing pipeline 300 may process packet each of the plurality of packets 401 of transmission 3200.

[0374] As shown in FIG. 32, in some embodiments, each of the transmitted packets 401 has a different (e.g., fixed or configurable) synchronization sequence 2802 transmitted the same sub-channel for each packet. As also shown in FIG. 32, in some embodiments, the hopping sequence 2804 for each of the 401 packets changes depending on content of synchronization sequence 2802.

[0375] In some embodiments, each of synchronization sequence 2802a, 2802b, 2802c, and 2802d, include one or more bits causing a selection of different coefficients (e.g., LCGa and LCGc, e.g., according FIGS. 29A and 29B) which causes a different hopping sequence 2804 to be used for associated packet 401.

[0376] FIG. 33 shows transmission 3300 of a plurality of packets 401, according to an embodiment of the present disclosure. Processing pipeline 300 may process packet each of the plurality of packets 401 of transmission 3300.

[0377] As shown in FIG. 33, in some embodiments, each of the transmitted packets 401 has a different (e.g., fixed or configurable) synchronization sequence 2802 transmitted in different sub-channel for each packet. As also shown in FIG. 33, in some embodiments, the hopping sequence 2804 for each of the 401 packets may be different and may be determined based on STFsubcarrier and/or content of synchronization sequence 2802.

[0378] In some embodiments, device 202 may use a first (e.g., fixed or configurable) STFsubcarrier (e.g., as shown in FIGS. 30-33) to transmit data to device 252 and may listen for a synchronization sequence from device 252 in a second (e.g., fixed or configurable, e.g., as shown in FIGS. 30-33). In some embodiments, the first and second STFsubcarriers may be the same.

[0379] FIG. 34 shows a block diagram of modulator 3400, according to an embodiment of the present disclosure. Modulator 302 may be implemented as modulator 3400. Modulator 3400 illustrates a possible implementation of processing pipeline 400.

[0380] As shown in FIG. 34, modulator 3400 includes scrambler block 402, FEC encoder block 404a and 404b, interleaver blocks 406a and 406b, DSSS blocks 408a, 408b, 408c, and 408d, bits to symbol blocks 410a and 410b, single sub-carrier mapping blocks 412a and 412b, inverse transform block 414, cyclic prefix block 416 and filtering block 418.

[0381] As shown in FIG. 34, in some embodiments, some of the processing blocks may be implemented with multiple instances (e.g., DSSS block 408 may have 4 instances,

which may advantageously allow for parallel processing). In some embodiments, a single instance of block (e.g., a single DSSS block 408) may be implemented, which may process data sequentially, e.g., by multiplexing among the different inputs, and possibly switching configuration of the block, e.g., based on the data being processed (e.g., using DSSS value of 2 for processing LTF field 2504, DSSS value of 6 for PHR Field 2506, and a DSSS value selected based on content of PHR field 2506 to process PHY payload field 2508).

[0382] As shown in FIG. 34, modulator 3400 may process different portions of packet 401 (e.g., in the form of PPDU 2500) in different manners.

[0383] For example, in some embodiments (as shown in FIG. 34), all content of STF field 2502 and LTF field 2504 is processed using DSSS block 408d and 408c, respectively (e.g., without scrambling, FEC encoding, or interleaving) to generate respective streams of chips. All content of PHR field 2506 is processed using FEC encoder block 404b, followed by interleaver block 406b, followed by DSSS block 408b (e.g., without scrambling) to generate a respective stream of chips. PHY payload field 2508 is processed by scrambler block 402, follows FEC encoder block 404a, interleaver block 406a, and DSSS block 408a to generate a respective stream of chips.

[0384] The outputs of DSSS blocks 408a, 408b, and 408c is concatenated by concatenation block 3402, and the concatenated stream of chips is processed by bits-to-symbols block 410a to generate a stream of symbols that are then mapped to a single carrier (e.g., according to a hopping sequence 2804). The stream of chips generated by DSSS 408d is processed by bits-to-symbols block 410b to generate a stream of symbols that are then mapped to a single carrier (e.g., to a fixed sub-channel STFsubcarrier).

[0385] The outputs of single sub-carrier mapping blocks 412a and 412b is concatenated by concatenation block 3404 to generate a concatenated sequence of symbols that is then processed by inverse transform 414, cyclic prefix block 416 and filtering block 418 to generate TX modulated signal.

[0386] In some embodiments, modulator 2400 may implement additional blocks (not shown), may omit one or more of the blocks shown in FIG. 34, and/or processes different portions of PPDU 2500 in a different manner than shown in FIG. 34. For example, in some embodiments, one or more of blocks 402, 404a, 404b, 406a, 406b, 416, 418, 3402 and 3404 may be omitted from modulator 3400. As another example, some embodiments may include a scrambler block (not shown) to process PHR field 2506 prior to FEC encoder block 404b. Other implementations are also possible.

[0387] In some embodiments, FEC encoder blocks 404a and 404b have the same configuration (e.g., same coding rate and polynomial, such as convolutional encoder 1000). In some embodiments, FEC encoder blocks 404a and 404b each have a different configuration (e.g., may use different coding rate and/or polynomial).

[0388] In some embodiments, interleaver blocks 406a and 406b have the same configuration (e.g., interleave according to the same algorithm, such as implemented as interleaver 1100 according to Equation 2). In some embodiments, interleaver blocks 406a and 406b each have a different configuration (e.g., may interleave according to different equations).

[0389] In some embodiments, DSSS blocks 408a, 408b, 408c, and 408d have the same configuration (e.g., same

DSSS value and polarity). In some embodiments, one or more of DSSS blocks **408a**, **408b**, **408c**, and **408d** may have a different configuration e.g., may use a different DSSS value and/or polarity) than another of DSSS blocks **408a**, **408b**, **408c**, and **408d**. For example, in some embodiments, DSSS blocks **408d** and **408c** may use a fixed DSSS value of 2, DSSS block **408b** may use a fixed DSSS value of 6, and DSSS block **408a** may use a DSSS value selected from a set (e.g., that includes DSSS values of 2, 4, and 6) depending on content of PHR field **2506**. In some such embodiments, the polarity used by each of DSSS blocks **408a**, **408b**, **408c**, and **408d** may be equal to each other, may be different, or may change (e.g., in a deterministic manner) based on one or more factors.

[0390] In some embodiments, bits-to-symbols blocks **410a**, and **410b**, have the same configuration (e.g., use the same modulation scheme, such as BPSK, and the same symbol duration SymDur). In some embodiments, bits-to-symbols blocks **410a**, and **410b** each have a different configuration (e.g., bits-to-symbols blocks **410a** may use BPSK as the modulation scheme, while bits-to-symbols blocks **410b** may use a modulation scheme different from BPSK, such as PSK, GFSK, FSK, ASK, CSS, QAM, APSK, CPS, MSK or OOK, and/or may use different symbol durations SymDur).

[0391] In some embodiments, single sub-carrier mapping blocks **412a** and **412b** may have different configurations. For example, in some embodiments, single sub-carrier mapping blocks **412a** may have a CH_{sel} that changes according to a hopping sequence (e.g., based on STFsubcarrier, LCGa, and LCGc) while single sub-carrier mapping blocks **412a** may have a CH_{sel} that is fixed for the entire PPDU **2500**.

[0392] In some embodiments, modulator **3400** may receive, e.g., from an upper layer (e.g., MAC layer), e.g., some or all content of PHY payload field **2508**. For example, in some embodiments, modulator **3400** may receive from a MAC layer all content of PSDU field **2540** and may generate all content of PPDU tail field **2542** and PAD field **2544**. In some embodiments, modulator **3400** may generate all content of PHY payload field **2508**. For example, in some embodiments, modulator **3400** may generate all content of PHY payload field **2508** during a test mode. Other implementations are also possible.

[0393] In some embodiments, modulator **3400** may receive, e.g., from an upper layer (e.g., MAC layer), e.g., some or all content of PHR field **2506**. For example, in some embodiments, modulator **3400** may receive from a MAC layer, e.g., all content of data rate field **2520** and frame length field **2522**, and may generate all content for HCS field **2524** and tail field **2526**. In some embodiments, modulator **3400** may generate all content of PHR field **2506**. Other implementations are also possible.

[0394] In some embodiments, modulator **3400** may receive, e.g., from an upper layer (e.g., MAC layer), e.g., some or all content of STF field **2502** and/or LTF field **2504**. For example, in some embodiments, modulator **3400** may receive from a MAC layer one or more (e.g., of the last) bits of STF field **2502**, e.g., for selecting hopping sequence **2804**, and may determine content of LTF field **2504** based on such bits. In some embodiments, modulator **3400** may generate all content of STF field **2502** and/or LTF field **2504**. Other implementations are also possible.

[0395] FIG. 35 shows a block diagram of demodulator **3500**, according to an embodiment of the present disclosure.

Demodulator **304** may be implemented as demodulator **3500**. Demodulator **3500** illustrates a possible implementation of processing pipeline **500**.

[0396] In some embodiments, demodulator **3500** is configured to demodulate a modulated signal generated using modulator **3400**, e.g., processing different portions of the modulated signal in a different manner, e.g., so as to recover the PPDU (e.g., **2500**), e.g., as shown in FIG. 35.

[0397] As shown in FIG. 35, demodulator **3500** includes filtering block **502**, STF synchronization block **3502**, Fourier Transform block **504**, extraction-from-carrier blocks **506a** and **506b**, inverse DSSS blocks **508a** and **508b**, de-interleaver blocks **510a** and **510b**, symbols-to-metrics blocks **512a** and **512b**, FEC decoder blocks **514a** and **514b**, and de-scrambler block **516**, and splitters **3504** and **3506**.

[0398] As shown in FIG. 35, in some embodiments, some of the processing blocks may be implemented with multiple instances (e.g., inverse DSSS block **508** may have 2 instances, which may advantageously allow for parallel processing). In some embodiments, a single instance of block (e.g., a single inverse DSSS block **508**) may be implemented, which may process data sequentially, e.g., by multiplexing among the different inputs, and possibly switching configuration of the block, e.g., based on the data being processed (e.g., using DSSS value of 2 for processing LTF field **2504**, DSSS value of 6 for PHR Field **2506**, and a DSSS value selected based on content of PHR field **2506** to process PHY payload field **2508**).

[0399] As shown in FIG. 35, demodulator **3500** may process different portions of packet **401** (e.g., in the form of PPDU **2500**) in different manners.

[0400] In some embodiments, STF synchronization block **3502** receives a signal from antenna **308** (e.g., after filtering (e.g., **502**) or directly from an ADC (e.g., **330**)). STF synchronization block **3502** monitors the received signal (e.g., after filtering by filtering block **502**) for detection of an STF field (e.g., **2502**) of a PPDU (e.g., **2500**) in a particular subcarrier channel STFsubcarrier (e.g., which may be fixed, or may vary dynamically, e.g., for each packet, or according to a hopping sequence, for example).

[0401] Once synchronization block **3502** determines that a received sequence matches a predetermined STF value (e.g., such as according to FIG. 26), Fourier Transform block **504** may begin processing of the received packet. For example, in some embodiments, STF synchronization block **3502** may signal, upon an STF match, a time/sample location for Fourier Transform block **504** to begin performing an FFT on the received sampled (e.g., on the output of filtering block **502**).

[0402] As shown in FIG. 35, demodulator **3500** may process different portions of PPDU **2500** in a different manner (e.g., in a reverse/inverse manner as modulator **3400**). For example, splitter **3502** may direct symbols associated with STF field **2502** to be processed by extraction-from-carrier block **506b** (e.g., using CH_{sel} equal to the same STFsubcarrier used by single sub-carrier mapping block **412b**), followed by inverse DSSS block **508b** (e.g., using the same DSSS value and polarity used by DSSS block **408d**), while directing the rest of the symbols to be processed by extraction-from-carrier block **506a** (e.g., using CH_{sel} based on the same hopping sequence as single sub-carrier mapping block **412a**) followed by inverse DSSS block **508a** (e.g., using the same DSSS value and polarity used by DSSS

blocks **408a**, **408b**, and **408c**, depending on which of fields **2504**, **2506**, and **208** is being processed).

[0403] Splitter **3506** may direct symbols associated with PHY payload field **2508** to be processed by de-interleaver block **510a**, followed by symbols-to-bits metrics block **512a**, followed by FEC decoder block **514a**, followed by de-scrambler block **516**; while directing symbols associated with PHR field **2506** to be processed by de-interleaver block **510b**, followed by symbols-to-bits metrics block **512b**, followed by FEC decoder block **514b**.

[0404] In some embodiments, demodulator **3500** places the demodulated data bits (e.g., of STF field **2502** from the output of inverse DSSS block **508b**, LTF field **2504** from the output of splitter **3506**, PHR field **2506** from the output of FEC decoder **514b**, and/or PHY payload field **2508** from the output of de-scrambler block **516**) into a receiver (RX) first-in-first-out (FIFO) buffer for further processing/use (e.g. by the same controller/demodulator performing the demodulation or by another controller). In some embodiments, some of the fields (e.g., STF field **2502** and/or LTF field **2506**) may not be placed in the FIFO for further processing.

[0405] In some embodiments, such as in embodiments in which STF field **2502** is a fixed (e.g., predetermined) sequence (e.g., without any configurable bits), blocks **3504**, **506b** and **508b** may be omitted and data associated with STF field **2502** may not be directly recovered. Instead, in some such embodiments, STF synchronization block **3502** may implement some such functionality to generate a match and an indication of the start location for beginning processing by Fourier Transform block **504** may be sufficient to indirectly determine the value of STF field **2502**.

[0406] In some embodiments, the entire contents of STF field **2502** (including the sub-carrier used for transmission of STF field **2502**) is predetermined and known to STF synchronization block **2502** to be able to generate a match (performed packet detection) and/or frequency offset determination. In some embodiments, STF field **2502** may include additional bits (not used for packet detection and/or frequency offset determination), which may be used for other purposes, such as for selecting a hopping sequence (e.g., to be followed to receive the rest of packet **401**).

[0407] In some embodiments, only some of the bits (e.g., the last 2 bits) of STF field **2502** may be processed by blocks **506b** and **508b** (e.g., while ignoring the rest of the bits of STF field **2502**).

[0408] FIG. 36 shows a block diagram of demodulator **3600**, according to an embodiment of the present disclosure. Demodulator **304** may be implemented as demodulator **3600**.

[0409] Demodulator **3600** operates in a similar manner as demodulator **3400**. Demodulator **3600** illustrates a possible implementation of processing pipeline **600**.

[0410] FIG. 37 shows a flow chart of embodiment method **3700** for receiving a packet (e.g., **401**), according to an embodiment of the present disclosure. Method **3700** may be performed, e.g., by demodulator **304**, such as by demodulators **3500** or **3600**.

[0411] During step **3702**, a demodulator (e.g., **304**, **3500**, **3600**) monitors a predetermined sub-channel for detecting, e.g., a predetermined and fixed, STF sequence (e.g., using STF synchronization block **3502**).

[0412] During step **3704**, in response to detecting a match of the STF sequence, a start of the packet is detected (e.g., using STF synchronization block **3502**).

[0413] During step **3706**, a transition from the STF sequence to an LTF sequence is detected or determined (e.g., using STF synchronization block **3502**).

[0414] During step **3708**, coarse and fine frequency offset are determined based on the STF and LTF sequences (e.g., using STF synchronization block **3502**).

[0415] During step **3710**, the packet (e.g., the symbols of the packet) are de-rotated according to the estimated frequency offset (e.g., using STF synchronization block **3502**). For example, in some embodiments, the symbols of the packet are mixed in frequency so as to cancel the estimated frequency offset performed during the STF. The resulting symbols may have no significant frequency offset (e.g., are effectively baseband symbols).

[0416] During step **3712**, the channel index selection scheme for selecting CH_{sel} (e.g., according to hopping sequence **2804**) is triggered, and is used for selecting the sub-channel for symbol extraction. For example, in some embodiments, the index selection scheme uses LCGa and LCGc, as well as the STFsubcarrier for determining the hopping sequence.

[0417] During step **3714**, the symbol pairs are extracted from the channels selected during step **3712** (e.g., by using the selected CH_{sel} for blocks **506a** and **506b**), and are then further processed (e.g., as shown in FIGS. 35 and 36).

[0418] FIG. 38 shows a flow chart of embodiment method **3800** for generating a hopping sequence (e.g., **2804**), according to an embodiment of the present disclosure. Method **3800** may be performed, e.g., by modulator **304** (e.g., by single sub-carrier mapping block **412**), such as by demodulators **3500** or **3600**.

[0419] During step **3802**, a demodulator (e.g., **304**, **3400**, such as by block **412**) configures a subcarrier channel (e.g., STFsubcarrier) for transmission of a synchronization sequence (e.g., **3802**), and one or more coefficients (e.g., LCGa and LCGc). In some embodiments, the STFsubcarrier and/or the one or more coefficients are known a priori, such as received out-of-band, programmed by a semiconductor manufacturer or a user of the device (e.g., **202**, **252**), etc.

[0420] During step **3804**, a starting seed for determining the hopping sequence (e.g., **2804**) is selected based on the STFsubcarrier received during step **3802**. For example, if $S=\{s(k)\}$, $0 \leq k \leq M-1$, where $s(k) \in (0, 255)$ is a vector that contains all the seed values for corresponding hopping sequences, a second vector $J=\{j(k)\}$, $0 \leq k \leq M-1$, where

$$j(k) \in \left(-\frac{N_a}{2}, \frac{N_a}{2}\right),$$

N_a is the number of active tones (e.g., sub-channels used for transmission) that contains the sub-channel index to be used in that specific symbol, and the starting seed for generating the hopping sequence (e.g., **2804**) be $s(0)=s(1)=DFT_{sz} + STFsubcarrier$, the starting seed for the algorithm, with $j(0)=j(1)=STFsubcarrier$.

[0421] During step **3806**, a variable S_{tmp} is initialized. In some embodiments, S_{tmp} is initialized to be the seed value from the previous iteration $s(k-1)$.

[0422] During step **3808**, the variable S_{tmp} is updated using a linear congruential random generation based on the

one or more coefficients received during step **3802**. For example, in some embodiments, the variable S_{tmp} is updated according to Equation 6.

[**0423**] During step **3810**, the P LSBs from the variable S_{tmp} are extracted and stored in variable r_{tmp} . In some embodiments, P may be given by Equation 7. In some embodiments, extraction of the P LSBs may be performed according to Equation 8.

[**0424**] During step **3812**, r_{tmp} is compared against three conditions:

$$\begin{aligned} r_{tmp} &\geq DFT_{SZ} - \frac{N_a}{2}; & 1. \\ r_{tmp} &\leq DFT_{SZ} + \frac{N_a}{2}; & 2. \\ \text{and} \\ r_{tmp} &\neq DFT_{SZ}, & 3. \end{aligned}$$

where N_a is the number of data tones in a symbol. In some embodiments, when all 3 conditions are true, then $j(k)=r_{tmp}-DFT_{SZ}$ and $s(k)=s_{tmp}$ are set during step **3814**. Otherwise, a new iteration is performed until the three conditions are satisfied.

[**0425**] In some embodiments, vector J includes the sequence of sub-channels (e.g., $j(k)$ represents the next channel), and vector S is the seed vector.

[**0426**] By selecting the seed using a linear congruential random generated, e.g., as in method **3800**, some embodiments advantageously generate a hopping sequence that does not visit any channels before repeating. In some embodiments, once the sequence has hopped through all channels in the set, the sequence repeats.

[**0427**] By using the STFsubcarrier as a seed for generating the hopping sequence, e.g., as in method **3800**, some embodiments advantageously generate different sequences when using different channels for transmission of the synchronization sequence, where the different sequences may advantageously be orthogonal to each other, or that do not substantially interfere that with each other. By generating such different hopping sequences, some embodiments may advantageously allow for geographical coexistence of multiple networks without substantially impacting the transmission error rate. In some embodiments, such different hopping sequences are rotated versions of each other.

[**0428**] In some embodiments (e.g., as illustrated in FIG. 2), data may be transmitted between two devices (e.g., between devices **202** and **252**). For example, FIG. 39 shows a flow chart of embodiment method **3900** for packet exchange, according to an embodiment of the present disclosure.

[**0429**] During step **3902**, device **202a** transmit a (e.g., broadcast) packet (e.g., **401**, such as **2500**) using a predetermined sub-channel x_{ch} for transmitting STF field **2502**, and using a predetermined hopping sequence (e.g., based on predetermined x_{LCGa} , x_{LCGc} and x_{ch} , e.g., using **3800**). In some embodiments, PHY payload **2508** of the broadcast packet may include a command/request for data.

[**0430**] During steps **3920**, device **252a** monitor the predetermined sub-channel x_{ch} for packet detection (e.g., using respective STF synchronization blocks **3502**). Upon detection of the broadcast packet (e.g., and responsive to the command/request for data), device **252a** performs carrier

sense (e.g., listen before talk) to transmit a packet (e.g., **401**, such as **2500**), which may be an acknowledgement (ACK) packet or a data packet, and may include data (e.g., such as sensed data).

[**0431**] Once device **252a** detects that sub-channel x_{ch} is idle, device **252a** begins transmission of a (e.g., unicast) packet during step **3924**, using the same sub-channel x_{ch} and the same hopping sequence during step **4002**.

[**0432**] During step **3904**, device **202a** listen for packets from device **252a** (e.g., monitoring sub-channel x_{ch} for STF detection).

[**0433**] This process may repeat periodically, as shown in FIG. 39.

[**0434**] In some embodiments, method **3900** may be synchronous. In some embodiments, method **3900** may be asynchronous. For example, in some embodiments, step **3902**, and **3920** may be omitted, and device **252a** may asynchronously transmit packets (e.g., as data becomes available) while device **202a** is (e.g., continuously) listening during step **3904**.

[**0435**] In the embodiment of FIG. 39, devices **202a** and **252** each use the same STFsubcarrier and same hopping sequence. In some embodiments, devices **202a** and **252** may transmit using different STFsubcarriers and/or hopping sequences. For example, FIG. 40 shows a flow chart of embodiment method **4000** for packet exchange, according to an embodiment of the present disclosure. Method **4000** is similar to method **3900**. In method **4000**, however, device **252a** transmit the packet during step **4024** using an STFsubcarrier equal to y_{ch} , which is different from the STFsubcarrier used by device **202a** during step **3902**. Device **252a** may also use a different hopping sequence (e.g., y_{LCGa} and y_{LGCc} may be different than x_{LCGa} and x_{LGCc}). As such, device **202a** listens for packets during step **4004** monitoring sub-channel y_{ch} and using the hopping sequence corresponding to step **4024**.

[**0436**] Some embodiments may transmit data from one device to multiple devices, from multiple devices to one device, or from multiple devices to multiple devices, e.g., within the same geographical area. For example, FIG. 41 illustrates communication system **4100**, according to an embodiment of the present disclosure. As shown in FIG. 41, communication system **4100**, includes multiple devices **252** (only 2 shown in FIG. 41) and a single device **202** as part of network **4110**.

[**0437**] In the embodiment of FIG. 41, all devices **252** communicate with device **202** but may not communicate with each other.

[**0438**] In some embodiments, all packet exchanges (e.g., **401**) between device **202** and devices **252a** and **252b** use the same fixed sub-channel (e.g., STFsubcarrier) for transmitting the synchronization sequence (e.g., **2802**). Thus, in some embodiments, device **202a** may transmit (e.g., broadcast) a synchronization sequence (e.g., **2802**), e.g., periodically (e.g., for each packet, as STF field **2502**) using a predetermined STFsubcarrier (e.g., sub-channel **9**), e.g., to all devices **252**; and each of devices **252** (e.g., **252a** and **252b**) may transmit (e.g., unicast) data (e.g., synchronously or asynchronously, sequentially, or simultaneously) to device **202a** using the same fixed sub-channel (e.g., sub-channel **9**) for transmitting synchronization sequence **2802** (e.g., for transmitting STF field **2502**). In some such embodiments, hopping sequences **2804** transmitted by each of the devices (e.g., **202a**, **252a**, and **252b**) may be the same

for packet (e.g., 401, 2800) transmissions. In some such embodiments, carrier sense or other listen-before-talk mechanism may be performed before attempting to transmit a packet (e.g., to avoid collisions).

[0439] FIG. 42 shows a flow chart of embodiment method 4200 for packet exchange, according to an embodiment of the present disclosure. Method 4200 is similar to method 3900. Method 4200, however, includes multiple devices 252a (e.g., as in network 4110).

[0440] Steps 3902, 3920, 3922, and 3924 may be performed in a similar or identical manner as described with respect to FIG. 39. In method 4200, however, device 252b also monitors during step 4240 the predetermined sub-channel x_{ch} for packet detection (e.g., using respective STF synchronization blocks 3502). Upon detection of the broadcast packet (e.g., and responsive to the command/request for data), each of devices 252a and 252b perform carrier sense (e.g., listen before talk) to transmit respective packets (e.g., 401, such as 2500) during steps 3922 and 4242, respectively.

[0441] In the example of FIG. 42, device 252a begins transmission of the unicast packet before device 252b during step 3924, using the same sub-channel x_{ch} and the same hopping sequence during step 3902. Since the sub-channel (x_{ch}) is busy for device 252b to transmit STF, device 252b waits until transmission during step 3924 is finalizes, and then begins transmission of a packet during step 4244 using the same sub-channel x_{ch} and the same hopping sequence during step 3902.

[0442] During step 4204, device 202a listen for packets from devices 252a and 252b. In some embodiments, device 202a may identify the origin/source of each packet based on an id field of each packet (e.g., in respective PHY payload fields 2508).

[0443] In some embodiments, method 4200 may be synchronous. In some embodiments, method 4200 may be asynchronous. For example, in some embodiments, step 3902, 3920, and 4240 may be omitted, and devices 252a and 252b may asynchronously transmit packets (e.g., as data becomes available) while device 202a is (e.g., continuously) listening during step 4204.

[0444] In some embodiments, the hopping sequences 2804 used by each of devices 252a and 252b may be different, e.g., when using the same fixed sub-channel for synchronization sequence 2802 (e.g., based on unique LCGa and/or LCGc values) or when using different channels for synchronization sequences 2802 (e.g., based on unique STFsubcarrier, with same or different coefficients LCGa and/or LCGc). For example, FIG. 43 shows a flow chart of embodiment method 4300 for packet exchange, according to an embodiment of the present disclosure. Method 4300 is similar to method 4200. Method 4300, however, includes multiple devices 252 (e.g., 252a and 252b) transmitting simultaneously.

[0445] Steps 3902, 3920, 4240, 3922, and 4242 may be performed in a similar or identical manner as described with respect to FIG. 42. In method 4200, however, device 252b begins transmission of a packet during step 4344 which at least partially overlaps with a packet transmission of device 252a during step 4324.

[0446] During step 4324, device 252a detects sub-channel x_{ch} as idle, and transmits synchronization sequence (e.g., 2802, e.g., STF field 2502) in sub-channel x_{ch} . Device 252b detects sub-channel x_{ch} as busy during step 4342 and continues to monitor sub-channel x_{ch} until sub-channel x_{ch}

becomes idle (e.g., once transmission by device 252a of its synchronization sequence finishes). Once sub-channel x_{ch} becomes idle, device 252b begins transmission of a packet during step 4344 using a different hopping sequence than device 252a during step 4324 (e.g., (y_{LCG_a}, y_{LCG_c}) may be different than (z_{LCG_a}, z_{LCG_c})).

[0447] By using different hopping sequences, some embodiments advantageously allow for less wait time for a device to begin transmission (e.g., carrier sense step 4342 may be shorter than carrier sense step 4242).

[0448] In some embodiments, 4344 may be performed simultaneously with step 4324, e.g., when using different sub-channel for transmission of respective synchronization sequences.

[0449] In some embodiments, step 4344 may use the same hopping sequence as step 4324, e.g., by starting transmission a delay time after transmission during step 4324, which may result in a rotated hopping sequence (e.g., to the offset in start times).

[0450] FIG. 44 illustrates symbol transmission of communication system 3900, according to an embodiment of the present disclosure. In particular, FIG. 44 illustrates data symbols transmitted from device 252a, data symbols transmitted from device 252b, wideband burst interference 4402, and narrowband interference 4404.

[0451] In some embodiments, device 202a may advantageously recover the data bits from the data symbols received from device 252a despite the presence of wideband burst interference 4402, narrowband interference 4404, and interferer symbols (e.g., symbols from device 252b, as well as possibly from other devices 252 (not shown)).

[0452] Similarly, in some embodiments (e.g., as in method 4300), device 202a may, in addition to recovering the data bits from the data symbols received from device 252a, may advantageously recover (e.g., in parallel) the data bits from the data symbols received from device 252b.

[0453] As can be seen in FIG. 44, narrowband interference 4404 may correspond to a transmission of a synchronization sequence 2804 (e.g., during step 4344), thereby illustrating coexistence of transmission from multiple devices without (e.g., substantially) degrading performance of data transmission.

[0454] As shown in FIG. 44, using different hopping sequences (e.g., due to rotation, or otherwise) may advantageously aid in avoiding collisions between simultaneous or partially overlapping in time transmissions (e.g., from multiple devices 252 to device 202).

[0455] FIG. 45 illustrates communication system 4500, according to an embodiment of the present disclosure. Communication system 4500 operates in a similar manner as communication 200. Communication system 4500, however, includes multiple networks (4510, 4512) overlapping in the same geographical location. Although only 2 devices are shown in FIG. 45 for each of the networks, each network may include more than 2 devices. In some embodiments, a device (e.g., 202 or 252) may belong to more than 1 network. Other implementations are also possible.

[0456] As shown in FIG. 45, each device 252 may communicate with a respective device 202 in a respective network. In some such embodiments, devices may not communicate across network (e.g., device 202a does not communicate with device 252b, and device 202b does not communicate with device 202a).

[0457] Packet exchanges in each of networks **4510** and **4512** may be performed, e.g., in a similar manner as in methods **3900**, **4000**, **4200**, or **4300**.

[0458] As can be seen in FIG. **44**, when networks overlap in the same geographical locations, (as illustrated in FIG. **45**), devices (e.g., **202**, **252**) may still successfully transmit and receive packets, e.g., in the presence of other transmissions and/or interference. For example, in some embodiments, the communication between device **202a** and device **252a** (and similarly between device **202b** and device **252b**) enjoys the same capability of recovering the data bits from the received symbols despite the presence of burst interference, narrowband interference, and interferer symbols (e.g., in a similar manner as illustrated in FIG. **44**).

[0459] In some embodiments, each communication between a device **202** and a respective device **252** in each of the networks (e.g., **4510**, **4512**) may use a different sub-channel for synchronization sequence **2802**. Using a unique sub-channel for transmitting synchronization sequence **2802** may advantageously allow for multiple networks to coexist in the same geographical location with low risk of collision.

[0460] FIG. **46** illustrates communication system **4600**, according to an embodiment of the present disclosure. Communication system **4600** operates in a similar manner as communication **4500**. Communication system **4600**, however, includes a device (e.g., **252a**) in multiple networks (e.g., **4610**, **4612**).

[0461] In some embodiments, each network may be configured in different manner (may use same or different STF subcarrier, same or different LCGa and LCGc coefficients, same or different STF sequence, etc.).

[0462] In some embodiments, the actual synchronization sequence **2802** for each network (**4610** and **4612**) may be different (e.g., to identify each network). In some such embodiments, hopping sequences **2804** may be different for each network (e.g., **4610** and **4612**), even when using the same fixed sub-channel for synchronization sequence **2802** (e.g., based on unique LCGa and LCGc values).

[0463] Packet exchanges in each of networks **4610** and **4612** may be performed, e.g., in a similar manner as in methods **3900**, **4000**, **4200**, or **4300**.

[0464] In some embodiments, the communication between device **252a** and device **202a** (and similarly between devices **252a** and **202b**) enjoys the same capability of recovering the data bits from the received symbols despite the presence of burst interference, narrowband interference, and interferer symbols (e.g., in a similar manner as illustrated in FIG. **44**).

[0465] In some embodiments, a device may operate according to different examples, e.g., at different times. For example, in an embodiment, a device (e.g., **252a**) may operate using method **3900** or **4000** when in network **200**, according to method **4200** or **4300**. In some embodiments, a device (e.g., **252a**) may simultaneously be part of different networks (e.g., **4110**, **4610**, **4612**) while operating in the same geographical area of other networks (e.g., **4510**). Other implementations are possible.

[0466] In some embodiments, settings for any network of communication systems **200**, **3900**, **4100**, **4500**, and **4600** may be negotiated, e.g., using method **2400**.

[0467] Example embodiments of the present disclosure are summarized here. Other embodiments can also be understood from the entirety of the specification and the claims filed herein.

[0468] Example 1. A method including: transmitting, by a first device, a first synchronization sequence in a single synchronization channel of a plurality of channels; and after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0469] Example 2. The method of example 1, where the hopping sequence includes an initial channel for transmission of an initial symbol of the first data, where the initial channel of the hopping sequence is equal to the synchronization channel.

[0470] Example 3. The method of one of examples 1 or 2, where the plurality of channels includes N channels, N being a positive integer, where the hopping sequence hops through the N channels in first N hops of the hopping sequence.

[0471] Example 4. The method of one of examples 1 to 3, where the hopping sequence does not include consecutive equal channels.

[0472] Example 5. The method of one of examples 1 to 4, where the hopping sequence includes consecutive equal channels.

[0473] Example 6. The method of one of examples 1 to 5, where transmitting the first synchronization sequence and the first data includes transmitting the first synchronization sequence and the first data sequentially in an uninterrupted manner.

[0474] Example 7. The method of one of examples 1 to 6, where transmitting the first synchronization sequence includes transmitting a short training field (STF) of a first packet, and where transmitting the first data includes transmitting header and payload fields of the first packet.

[0475] Example 8. The method of one of examples 1 to 7, where transmitting the STF includes transmitting the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and where transmitting the header field includes transmitting the header field using DSSS with a second DSSS value different than the first DSSS value.

[0476] Example 9. The method of one of examples 1 to 8, where transmitting the payload field includes transmitting the payload field using DSSS with a third DSSS value that is based on content of the header field.

[0477] Example 10. The method of one of examples 1 to 9, where the second DSSS value is equal to 6, and the first DSSS value is equal to 2.

[0478] Example 11. The method of one of examples 1 to 10, where transmitting the first data includes transmitting a long training field (LTF) of the first packet before transmitting the header and payload fields.

[0479] Example 12. The method of one of examples 1 to 11, where transmitting the STF includes transmitting the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and where transmitting the LTF includes transmitting the LTF using DSSS with the first DSSS value.

[0480] Example 13. The method of one of examples 1 to 12, where the LTF includes 26 bits.

[0481] Example 14. The method of one of examples 1 to 13, further including: receiving second data from a MAC layer; and scrambling the second data to generate scrambled data, where transmitting the payload field includes transmitting the payload field based on the scrambled data.

[0482] Example 15. The method of one of examples 1 to 14, further including: generating a first sequence of mapped symbols including symbols of the STF; generating a second

sequence of mapped symbols including symbols of the header and payload fields; concatenating the first and second sequences of mapped symbols to generate a concatenated sequence; performing an inverse transform to the concatenated sequence to generate a sample stream; and applying cyclic prefix to the sample stream, where transmitting the first synchronization sequence and the first data includes transmitting the first synchronization sequence and the first data based on the sample stream.

[0483] Example 16. The method of one of examples 1 to 15, where the synchronization sequence includes a plurality of symbols of a first type, and where the first data includes a plurality of symbols of the first type.

[0484] Example 17. The method of one of examples 1 to 16, where the symbols of the first type are orthogonal frequency-division multiplexing (OFDM) symbols encoded using differential binary phase shift keying (DBPSK).

[0485] Example 18. The method of one of examples 1 to 17, where the synchronization sequence includes a plurality of symbols of a first type, and where the first data includes a plurality of symbols of a second type that is different from the first type.

[0486] Example 19. The method of one of examples 1 to 18, where the symbols of the second type are orthogonal frequency-division multiplexing (OFDM) symbols encoded using binary phase shift keying (BPSK).

[0487] Example 20. The method of one of examples 1 to 19, where the synchronization sequence is transmitted in pairs of symbols using direct sequence spread spectrum (DSSS).

[0488] Example 21. The method of one of examples 1 to 20, where each symbol of the synchronization sequence is a binary phase shift keying (BPSK) symbol.

[0489] Example 22. The method of one of examples 1 to 21, where the plurality of channels is a plurality of orthogonal frequency-division multiplexing (OFDM) channels.

[0490] Example 23. The method of one of examples 1 to 22, further including generating the hopping sequence based on the synchronization channel.

[0491] Example 24. The method of one of examples 1 to 23, further including generating the hopping sequence based on first and second coefficients.

[0492] Example 25. The method of one of examples 1 to 24, where: the first coefficient is selected from a set that includes: 17, 29, 37, 41, 53, 61, 73, or 89; and the second coefficient is selected from a set that includes 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, or 127.

[0493] Example 26. The method of one of examples 1 to 25, where generating the hopping sequence further includes generating the hopping sequence based on the synchronization channel.

[0494] Example 27. The method of one of examples 1 to 26, further including generating the hopping sequence using a linear congruential random generator.

[0495] Example 28. The method of one of examples 1 to 27, further including after transmitting the first data, monitoring, by the first device, the synchronization channel for detection of a second synchronization sequence.

[0496] Example 29. The method of one of examples 1 to 28, where the second synchronization sequence is equal to the first synchronization sequence.

[0497] Example 30. The method of one of examples 1 to 29, further including, responsive to detecting the second

synchronization sequence, processing, by the first device, a second packet that includes the second synchronization sequence.

[0498] Example 31. The method of one of examples 1 to 30, where processing the second packet includes processing a header field and a payload field of the second packet according to the hopping sequence.

[0499] Example 32. The method of one of examples 1 to 31, further including, after processing the second packet, transmitting, by the first device, a third synchronization sequence in the synchronization channel.

[0500] Example 33. The method of one of examples 1 to 32, further including: receiving the second synchronization sequence; and using the received second synchronization sequence for determining a frequency offset.

[0501] Example 34. The method of one of examples 1 to 33, where the plurality of channels consists of 12, 26, 54, or 104 channels.

[0502] Example 35. The method of one of examples 1 to 34, where none of the plurality of channels is a pilot channel for transmitting pilot waveforms.

[0503] Example 36. The method of one of examples 1 to 35, where the first synchronization sequence includes 160 chips.

[0504] Example 37. The method of one of examples 1 to 36, further including: waking up the first device from a sleep mode, where transmitting the first synchronization sequence includes transmitting the first synchronization channel after waking up from the sleep mode; and after transmitting the first synchronization sequence, transitioning the first device into the sleep mode.

[0505] Example 38. The method of one of examples 1 to 37, where the first data includes location data of the first device.

[0506] Example 39. The method of one of examples 1 to 38, further including tracking the first device based on the location data.

[0507] Example 40. The method of one of examples 1 to 39, further including collecting, by the first device, health data of an animal, where the first data includes the health data.

[0508] Example 41. The method of one of examples 1 to 40, collecting, by the first device, soil data, where the first data includes soil data.

[0509] Example 42. The method of one of examples 1 to 41, where the first data includes data indicative of whether a parking spot is being used.

[0510] Example 43. The method of one of examples 1 to 42, further including tracking a state of the parking spot based on the first data.

[0511] Example 44. The method of one of examples 1 to 43, where the first data includes metering data associated with electricity, water, or gas usage.

[0512] Example 45. A method including: monitoring, by a first device, a single synchronization channel of a plurality of channels for detection of a first synchronization sequence; and responsive to detecting the first synchronization sequence, extracting, by the first device, first data symbols associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0513] Example 46. The method of example 45, where the hopping sequence includes an initial channel for reception of

an initial symbol of the first data symbols, where the initial channel of the hopping sequence is equal to the synchronization channel.

[0514] Example 47. The method of one of examples 45 or 46, where the plurality of channels includes N channels, N being a positive integer, where the hopping sequence hops through the N channels in first N hops of the hopping sequence.

[0515] Example 48. The method of one of examples 45 to 47, where the hopping sequence does not include consecutive equal channels.

[0516] Example 49. The method of one of examples 45 to 48, where the hopping sequence includes consecutive equal channels.

[0517] Example 50. The method of one of examples 45 to 49, where receiving the first synchronization sequence and the first data symbols includes receiving the first synchronization sequence and the first data symbols sequentially in an uninterrupted manner.

[0518] Example 51. The method of one of examples 45 to 50, where receiving the first synchronization sequence includes receiving a short training field (STF) of a first packet, and where receiving the first data symbols includes receiving a header and payload fields of the first packet.

[0519] Example 52. The method of one of examples 45 to 51, where receiving the STF includes receiving the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and where receiving the header field includes receiving the header field using DSSS with a second DSSS value different than the first DSSS value, the method further including: determining bits of the STF based on the first DSSS value; and determining bits of the header field based on the second DSSS value.

[0520] Example 53. The method of one of examples 45 to 52, where receiving the payload field includes receiving the payload field using DSSS with a third DSSS value that is based on content of the header field.

[0521] Example 54. The method of one of examples 45 to 53, where the second DSSS value is equal to 6, and the first DSSS value is equal to 2.

[0522] Example 55. The method of one of examples 45 to 54, where receiving the first data symbols includes receiving a long training field (LTF) of the first packet before receiving the header and payload fields.

[0523] Example 56. The method of one of examples 45 to 55, where the LTF includes 26 symbols.

[0524] Example 57. The method of one of examples 45 to 56, where receiving the STF includes receiving the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and where receiving the LTF includes receiving the LTF using DSSS with the first DSSS value, the method further including: determining bits of the STF based on the first DSSS value; and determining bits of the LTF based on the first DSSS value.

[0525] Example 58. The method of one of examples 45 to 57, further including: performing a descrambling operation based on symbols of the payload field; determining bits of the payload field based on the descrambling operation; and determining bits of the header field without performing a descrambling operation.

[0526] Example 59. The method of one of examples 45 to 58, where the synchronization sequence includes a plurality of symbols of a first type, and where the first data symbols include a plurality of symbols of the first type.

[0527] Example 60. The method of one of examples 45 to 59, where the symbols of the first type are orthogonal frequency-division multiplexing (OFDM) symbols encoded using differential binary phase shift keying (DBPSK).

[0528] Example 61. The method of one of examples 45 to 60, where the synchronization sequence includes a plurality of symbols of a first type, and where the first data symbols include a plurality of symbols of a second type that is different from the first type.

[0529] Example 62. The method of one of examples 45 to 61, where the symbols of the second type are orthogonal frequency-division multiplexing (OFDM) symbols encoded using binary phase shift keying (BPSK).

[0530] Example 63. The method of one of examples 45 to 62, where the first synchronization sequence includes 160 symbols.

[0531] Example 64. The method of one of examples 45 to 63, where the synchronization sequence is in pairs of symbols using direct sequence spread spectrum (DSSS).

[0532] Example 65. The method of one of examples 45 to 64, where each symbol of the synchronization sequence is a binary phase shift keying (BPSK) symbol.

[0533] Example 66. The method of one of examples 45 to 65, where the plurality of channels is a plurality of orthogonal frequency-division multiplexing (OFDM) channels.

[0534] Example 67. The method of one of examples 45 to 66, further including generating the hopping sequence based on the synchronization channel.

[0535] Example 68. The method of one of examples 45 to 67, further including generating the hopping sequence based on first and second coefficients.

[0536] Example 69. The method of one of examples 45 to 68, where: the first coefficient is selected from a set that includes: 17, 29, 37, 41, 53, 61, 73, or 89; and the second coefficient is selected from a set that includes 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, or 127.

[0537] Example 70. The method of one of examples 45 to 69, where generating the hopping sequence further includes generating the hopping sequence based on the synchronization channel.

[0538] Example 71. The method of one of examples 45 to 70, further including generating the hopping sequence using a linear congruential random generator.

[0539] Example 72. The method of one of examples 45 to 71, further including, after receiving the first data symbols, transmitting, by the first device, a second synchronization sequence in the synchronization channel.

[0540] Example 73. The method of one of examples 45 to 72, where the second synchronization sequence is equal to the first synchronization sequence.

[0541] Example 74. The method of one of examples 45 to 73, further including, performing carrier sense on the synchronization channel, where transmitting the second synchronization sequence includes transmitting the second synchronization sequence responsive to determining that the synchronization channel is idle.

[0542] Example 75. The method of one of examples 45 to 74, further including, after transmitting the second synchronization sequence, monitoring, by the first device, the synchronization channel for detection of a third synchronization sequence.

[0543] Example 76. The method of one of examples 45 to 75, where the plurality of channels consists of 12, 26, 54, or 104 channels.

[0544] Example 77. The method of one of examples 45 to 76, where none of the plurality of channels is a pilot channel for transmitting pilot waveforms.

[0545] Example 78. The method of one of examples 45 to 77, further including: detecting a start location of a first packet, that includes the first synchronization sequence and the first data symbols, based on detecting the first synchronization sequence; and performing a Fourier transform on the first packet based on the start location.

[0546] Example 79. The method of one of examples 45 to 78, where the first synchronization sequence includes a short training field (STF), the method further including: detecting a transition from the STF to a long training field (LTF) of the first packet; estimating coarse and fine frequency offset based on the STF and LTF; de-rotating the first packet based on the estimated frequency offset; triggering channel selection based on the hopping sequence on the LTF; and extracting pairs symbols of the LTF responsive to triggering the channel selection.

[0547] Example 80. The method of one of examples 45 to 79, further including alternating, by the first device, between an active mode with a receive path of a transceiver of the first device enabled and a sleep mode with the receive path disabled, where monitoring the single synchronization channel for detection of the first synchronization sequence includes monitoring the single synchronization channel when the first device is the active mode.

[0548] Example 81. The method of one of examples 45 to 80, further including periodically monitoring the single synchronization channel.

[0549] Example 82. The method of one of examples 45 to 81, further including controlling a light based on the first data.

[0550] Example 83. A method including: generating, by a first device, a first plurality of spreads of chips corresponding to a plurality of bits of a synchronization sequence of a packet; generating, by the first device, a second plurality of spreads of chips corresponding to a plurality of bits of a header field of the packet; generating, by the first device, a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; generating, by the first device, a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; transmitting, by the first device, the first plurality of pairs of BPSK symbols in a single synchronization channel of a plurality of channels; and after transmitting the first plurality of pairs of BPSK symbols, transmitting, by the first device, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission at a time.

[0551] Example 84. The method of example 83, where the hopping sequence does not repeat a channel of the plurality of channels until all channels of the plurality of channels are used.

[0552] Example 85. The method of one of examples 83 or 84, where the hopping sequence is based on the channel used for transmitting the synchronization sequence.

[0553] Example 86. A device including: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single synchronization

channel of a plurality of channels, and after transmitting the first synchronization sequence, transmit first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0554] Example 87. The device of example 86, further including an antenna coupled to the transceiver, where transmitting the first synchronization sequence includes transmitting the first synchronization sequence via the antenna.

[0555] Example 88. A device including: a transceiver; and a controller configured to: monitor, using the transceiver, a single synchronization channel of a plurality of channels for detection of a first synchronization sequence, and responsive to detecting the first synchronization sequence, extract first data symbols associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.

[0556] Example 89. A device including: a transceiver; and a controller configured to: generate a first plurality of spreads of chips corresponding to a plurality of bits of a synchronization sequence of a packet, generate a second plurality of spreads of chips corresponding to a plurality of bits of a header field of the packet, generate a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips, generate a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips, transmit, via the transceiver, the first plurality of pairs of BPSK symbols in a single synchronization channel of a plurality of channels, and after transmitting the first plurality of pairs of BPSK symbols, transmit, via the transceiver, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission at a time.

[0557] Example 90. A method including: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channel; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; after transmitting the first data, transmitting, by the first device, a second synchronization sequence in a single second synchronization channel of the plurality of channels; and after transmitting the second synchronization sequence, transmitting, by the first device, second data associated with the second synchronization sequence according to a second hopping sequence using a single channel of the plurality of channels at a time, where: the first synchronization channel is different from the second synchronization channel, or the first synchronization sequence is different from the second synchronization sequence, or the second hopping sequence is different from the first hopping sequence.

[0558] Example 91. The method of example 90, where the first synchronization channel is different from the second synchronization channel.

[0559] Example 92. The method of one of examples 90 or 91, further including: generating the first hopping sequence based on first synchronization channel; and generating the second hopping sequence based on second synchronization channel.

[0560] Example 93. The method of one of examples 90 to 92, where the first synchronization sequence is different from the second synchronization sequence.

[0561] Example 94. The method of one of examples 90 to 93, where the first synchronization sequence is equal to the second synchronization sequence.

[0562] Example 95. The method of one of examples 90 to 94, where the second hopping sequence is different from the first hopping sequence.

[0563] Example 96. The method of one of examples 90 to 95, where the first synchronization sequence is different from the second synchronization sequence.

[0564] Example 97. The method of one of examples 90 to 96, further including: generating the first hopping sequence based on a first coefficient, where the first coefficient is based on content of the first synchronization sequence; and generating the second hopping sequence based on a second coefficient, where the second coefficient is based on content of the second synchronization sequence.

[0565] Example 98. The method of one of examples 90 to 97, where the first synchronization channel is equal to the second synchronization channel.

[0566] Example 99. The method of one of examples 90 to 98, where the second hopping sequence is different from the first hopping sequence.

[0567] Example 100. The method of one of examples 90 to 99, where the first data is directed to a second device, and where the second data is directed to a third device.

[0568] Example 101. The method of one of examples 90 to 100, where transmitting the first synchronization sequence and the first data includes broadcasting a first packet that includes the first synchronization sequence and the first data.

[0569] Example 102. The method of one of examples 90 to 101, where the first hopping sequence includes an initial channel for transmission of an initial symbol of the first data, where the initial channel of the first hopping sequence is equal to the first synchronization channel.

[0570] Example 103. The method of one of examples 90 to 102, where the plurality of channels includes N channels, N being a positive integer, where the first hopping sequence hops through the N channels in first N hops of the first hopping sequence.

[0571] Example 104. The method of one of examples 90 to 103, where the first hopping sequence does not include consecutive equal channels.

[0572] Example 105. The method of one of examples 90 to 104, where the first hopping sequence includes consecutive equal channels.

[0573] Example 106. The method of one of examples 90 to 105, where transmitting the first synchronization sequence includes transmitting a short training field (STF) of a first packet, and where transmitting the first data includes transmitting header and payload fields of the first packet.

[0574] Example 107. The method of one of examples 90 to 106, further including: generating a first sequence of mapped symbols including symbols of the STF; generating a second sequence of mapped symbols including symbols of the header and payload fields; concatenating the first and second sequences of mapped symbols to generate a concatenated sequence; performing an inverse transform to the concatenated sequence to generate a sample stream; and applying cyclic prefix using windowing to the sample stream, where transmitting the first synchronization

sequence and the first data includes transmitting the first synchronization sequence and the first data based on the sample stream.

[0575] Example 108. The method of one of examples 90 to 107, where the synchronization sequence includes a plurality of symbols of a first type, and where the first data includes a plurality of symbols of a second type that is different from the first type.

[0576] Example 109. The method of one of examples 90 to 108, where the symbols of the second type are orthogonal frequency-division multiplexing (OFDM) symbols encoded using binary phase shift keying (BPSK).

[0577] Example 110. A method including: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data, monitoring, by the first device, a single second synchronization channel for detection of a second synchronization sequence, where the first synchronization channel is different from the second synchronization channel.

[0578] Example 111. The method of example 110, where transmitting the first synchronization sequence and the first data includes broadcasting a first packet that includes the first synchronization sequence and the first data.

[0579] Example 112. The method of one of examples 110 or 111, further including, after transmitting the first data, monitoring, by the first device, a single third synchronization channel for detection of a third synchronization sequence, where the third synchronization channel is different from the first and second synchronization channel, where the second synchronization channel is associated with a second device, and where the third synchronization channel is associated with a third device.

[0580] Example 113. A method including: generating a plurality of spreads of chips corresponding to a plurality of bits of a first packet; generating a plurality of binary phase shift keying (BPSK) symbols based on the plurality of spreads of chips; repeating a portion of each respective BPSK symbol of the plurality of BPSK symbols before the respective BPSK symbol; applying a windowing filter during a first portion of the repeated portion of each BPSK symbol to generate a second plurality of BPSK symbols; and transmitting the second plurality of BPSK symbols, as pairs of symbols, in respective channels of a plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission of the second plurality of BPSK symbols at a time.

[0581] Example 114. The method of example 113, where the hopping sequence does not repeat a channel of the plurality of channels until all channels of the plurality of channels are used.

[0582] Example 115. The method of one of examples 113 or 114, further including transmitting a synchronization sequence in one of the plurality of channels before transmitting the second plurality of BPSK symbols according to the hopping sequence.

[0583] Example 116. The method of one of examples 113 to 115, where the hopping sequence is based on the one channel used for transmitting the synchronization sequence.

[0584] Example 117. The method of one of examples 113 to 116, where the first portion of the repeated portion is half a portion of the repeated portion.

[0585] Example 118. The method of one of examples 113 to 117, where the windowing filter performs a linear windowing function that combines samples of a previous symbol together with samples of the repeated portion of the symbol.

[0586] Example 119. A method including: transmitting, by a first device, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data: monitoring, by the first device, a single second synchronization channel for detection of a second synchronization sequence associated with a second device, and monitoring, by the first device, the single second synchronization channel for detection of a third synchronization sequence associated with a third device.

[0587] Example 120. The method of example 119, where the second synchronization sequence is different from the third synchronization sequence.

[0588] Example 121. The method of one of examples 119 or 120, where the second synchronization sequence is equal to the third synchronization sequence.

[0589] Example 122. The method of one of examples 119 to 121, where the first synchronization channel is different from the second synchronization channel.

[0590] Example 123. The method of one of examples 119 to 122, where the first synchronization channel is equal to the second synchronization channel.

[0591] Example 124. The method of one of examples 119 to 123, further including: receiving, by the first device, the second synchronization sequence in the second synchronization channel, followed by second data according to a second hopping sequence; and receiving, by the first device, the third synchronization sequence in the third synchronization channel, followed by third data according to a third hopping sequence.

[0592] Example 125. The method of one of examples 119 to 124, where receiving the second data according to the second hopping sequence partially overlaps in time with receiving the third data according to the third hopping sequence.

[0593] Example 126. A method including: generating a first plurality of spreads of chips corresponding to a plurality of bits of a first packet; generating a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; transmitting the first plurality of pairs of BPSK symbols in respective channels of a plurality of channels according to a first hopping sequence; generating a second plurality of spreads of chips corresponding to a plurality of bits of a second packet; generating a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; and transmitting the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a second hopping sequence, where the second hopping sequence is different from the first hopping sequence, and where only a single channel of the plurality of channels is used for transmission at a time.

[0594] Example 127. The method of example 126, where the hopping sequence does not repeat a channel of the plurality of channels until all channels of the plurality of channels are used.

[0595] Example 128. The method of one of examples 126 or 127, further including transmitting a first synchronization sequence in one of the plurality of channels before transmitting the first plurality of pairs of BPSK symbols according to the first hopping sequence.

[0596] Example 129. The method of one of examples 126 to 128, where the first hopping sequence is based on the one channel used for transmitting the first synchronization sequence.

[0597] Example 130. A method including: negotiating, by a first device with a second device, a first symbol duration; transmitting, by the first device, a first synchronization sequence in a single synchronization channel of a plurality of channels using the negotiated first symbol duration; and after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time using the negotiated symbol duration.

[0598] Example 131. The method of example 130, where negotiating the symbol duration includes transmitting, by the first device, a packet using a second symbol duration different from the first symbol duration.

[0599] Example 132. A device including: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; after transmitting the first data, transmit, via the transceiver, a second synchronization sequence in a single second synchronization channel of the plurality of channels; and after transmitting the second synchronization sequence, transmit, via the transceiver, second data associated with the second synchronization sequence according to a second hopping sequence using a single channel of the plurality of channels at a time, where: the first synchronization channel is different from the second synchronization channel, or the first synchronization sequence is different from the second synchronization sequence, or the second hopping sequence is different from the first hopping sequence.

[0600] Example 133. A device including: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data, monitor a single second synchronization channel for detection of a second synchronization sequence, where the first synchronization channel is different from the second synchronization channel.

[0601] Example 134. A device including: a transceiver; and a controller configured to: generate a plurality of spreads of chips corresponding to a plurality of bits of a first packet; generate a plurality of binary phase shift keying (BPSK) symbols based on the plurality of spreads of chips; repeat a

portion of each respective BPSK symbol of the plurality of BPSK symbols before the respective BPSK symbol; apply a windowing filter during a first portion of the repeated portion of each BPSK symbol to generate a second plurality of BPSK symbols; and transmit, via the transceiver, the second plurality of BPSK symbols, as pairs of symbols, in respective channels of a plurality of channels according to a hopping sequence, where only a single channel of the plurality of channels is used for transmission of the second plurality of BPSK symbols at a time.

[0602] Example 135. A device including: a transceiver; and a controller configured to: transmit, via the transceiver, a first synchronization sequence in a single first synchronization channel of a plurality of channels; after transmitting the first synchronization sequence, transmitting, transmit, via the transceiver, first data associated with the first synchronization sequence according to a first hopping sequence using a single channel of the plurality of channels at a time; and after transmitting the first data: monitor, using the transceiver, a single second synchronization channel for detection of a second synchronization sequence associated with a second device, and monitor, using the transceiver, the single second synchronization channel for detection of a third synchronization sequence associated with a third device.

[0603] Example 136. A device including: a transceiver; and a controller configured to: generate a first plurality of spreads of chips corresponding to a plurality of bits of a first packet; generate a first plurality of pairs of binary phase shift keying (BPSK) symbols based on the first plurality of spreads of chips; transmit, via the transceiver, the first plurality of pairs of BPSK symbols in respective channels of a plurality of channels according to a first hopping sequence; generate a second plurality of spreads of chips corresponding to a plurality of bits of a second packet; generate a second plurality of pairs of BPSK symbols based on the second plurality of spreads of chips; and transmit, via the transceiver, the second plurality of pairs of BPSK symbols in respective channels of the plurality of channels according to a second hopping sequence, where the second hopping sequence is different from the first hopping sequence, and where only a single channel of the plurality of channels is used for transmission at a time.

[0604] Example 137. A device including: a transceiver; and a controller configured to: negotiate, via the transceiver, with a second device a first symbol duration; transmit, via the transceiver, a first synchronization sequence in a single synchronization channel of a plurality of channels using the negotiated first symbol duration; and after transmitting the first synchronization sequence, transmit, via the transceiver, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time using the negotiated symbol duration.

[0605] While this disclosure has been described with reference to illustrative embodiments, this description is not limiting. Various modifications and combinations of the illustrative embodiments, as well as other embodiments, will be apparent to persons skilled in the art upon reference to the description.

What is claimed is:

1. A method comprising:
transmitting, by a first device, a first synchronization sequence in a single synchronization channel of a plurality of channels; and
after transmitting the first synchronization sequence, transmitting, by the first device, first data associated with the first synchronization sequence according to a hopping sequence using a single channel of the plurality of channels at a time.
2. The method of claim 1, wherein the hopping sequence comprises an initial channel for transmission of an initial symbol of the first data, wherein the initial channel of the hopping sequence is equal to the synchronization channel.
3. The method of claim 1, wherein the plurality of channels includes N channels, N being a positive integer, wherein the hopping sequence hops through the N channels in first N hops of the hopping sequence.
4. The method of claim 1, wherein the hopping sequence does not include consecutive equal channels.
5. The method of claim 1, wherein the hopping sequence includes consecutive equal channels.
6. The method of claim 1, wherein transmitting the first synchronization sequence and the first data comprises transmitting the first synchronization sequence and the first data sequentially in an uninterrupted manner.
7. The method of claim 1, wherein transmitting the first synchronization sequence comprises transmitting a short training field (STF) of a first packet, and wherein transmitting the first data comprises transmitting header and payload fields of the first packet.
8. The method of claim 7, wherein transmitting the STF comprises transmitting the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and wherein transmitting the header field comprises transmitting the header field using DSSS with a second DSSS value different than the first DSSS value.
9. The method of claim 8, wherein transmitting the payload field comprises transmitting the payload field using DSSS with a third DSSS value that is based on content of the header field.
10. The method of claim 7, wherein transmitting the first data comprises transmitting a long training field (LTF) of the first packet before transmitting the header and payload fields.
11. The method of claim 10, wherein transmitting the STF comprises transmitting the STF using direct sequence spread spectrum (DSSS) with a first DSSS value, and wherein transmitting the LTF comprises transmitting the LTF using DSSS with the first DSSS value.
12. The method of claim 7, further comprising:
receiving second data from a MAC layer; and
scrambling the second data to generate scrambled data,
wherein transmitting the payload field comprises transmitting the payload field based on the scrambled data.
13. The method of claim 7, further comprising:
generating a first sequence of mapped symbols including symbols of the STF;
generating a second sequence of mapped symbols including symbols of the header and payload fields;
concatenating the first and second sequences of mapped symbols to generate a concatenated sequence;
performing an inverse transform to the concatenated sequence to generate a sample stream; and
applying cyclic prefix to the sample stream, wherein transmitting the first synchronization sequence and the

first data comprises transmitting the first synchronization sequence and the first data based on the sample stream.

14. The method of claim 1, wherein the synchronization sequence comprises a plurality of symbols of a first type, and wherein the first data comprises a plurality of symbols of the first type.

15. The method of claim 1, wherein the synchronization sequence comprises a plurality of symbols of a first type, and wherein the first data comprises a plurality of symbols of a second type that is different from the first type.

16. The method of claim 1, wherein the synchronization sequence is transmitted in pairs of symbols using direct sequence spread spectrum (DSSS), wherein each symbol of the synchronization sequence is a binary phase shift keying (BPSK) symbol, and wherein the plurality of channels is a plurality of orthogonal frequency-division multiplexing (OFDM) channels.

17. The method of claim 1, further comprising generating the hopping sequence based on the synchronization channel.

18. The method of claim 1, further comprising generating the hopping sequence based on first and second coefficients, wherein:

the first coefficient is selected from a set that includes: 17, 29, 37, 41, 53, 61, 73, or 89; and

the second coefficient is selected from a set that includes 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, or 127.

19. The method of claim 1, further comprising after transmitting the first data, monitoring, by the first device, the synchronization channel for detection of a second synchronization sequence, wherein the second synchronization sequence is equal to the first synchronization sequence.

20. The method of claim 1, wherein none of the plurality of channels is a pilot channel for transmitting pilot waveforms.

21. The method of claim 1, further comprising:
waking up the first device from a sleep mode, wherein
transmitting the first synchronization sequence comprises transmitting the first synchronization channel after waking up from the sleep mode; and
after transmitting the first synchronization sequence, transitioning the first device into the sleep mode.

* * * * *