METHOD AND APPARATUS FOR TRANSMITTING PHYSICAL LAYER PROTOCOL DATA UNIT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to Chinese Patent Application No. 202010769451.7, filed with the China National Intellectual Property Administration on August 3, 2020 and entitled "METHOD AND APPARATUS FOR TRANSMITTING PHYSICAL LAYER PROTOCOL DATA UNIT", which is incorporated herein by reference in its entirety; this application claims priority to Chinese Patent Application No. 202010851834.9, filed with the China National Intellectual Property Administration on August 21, 2020 and entitled "METHOD AND APPARATUS FOR TRANSMITTING PHYSICAL LAYER PROTOCOL DATA UNIT", which is incorporated herein by reference in its entirety; and this application claims priority to Chinese Patent Application No. 202010930892.0, filed with the China National Intellectual Property Administration on September 7, 2020 and entitled "METHOD AND APPARATUS FOR TRANSMITTING PHYSICAL LAYER PROTOCOL DATA UNIT", which is incorporated herein by reference in its entirety.

~~**[0001]**~~**[0002]**

TECHNICAL FIELD

~~**[0002]**~~**[0003]** Embodiments of this application relate to the field of wireless communication technologies, and in particular, to a method and an apparatus for transmitting a physical layer protocol data unit.

BACKGROUND

~~**[0003]**~~**[0004]** Because performance of a wireless communications system is greatly affected by a radio channel, such as shadow fading and frequency selective fading, a propagation path between a transmitter and a receiver is very complicated. Therefore, channel estimation needs to

be performed in coherent detection of an orthogonal frequency division multiplexing (orthogonal frequency division multiplexing, OFDM) system. Channel estimation is a process of estimating, under a specific criterion, a parameter of a channel through which a radio signal passes. Accuracy of channel estimation directly affects the performance of the whole system.

~~[0004]~~[0005] Therefore, in wireless local area network (wireless local area network, WLAN) standards that use an OFDM technology as a core, such as Institute of Electrical and Electronics Engineers (institute of electrical and electronics engineers, IEEE) 802.11g/a, 802.11n, and 802.11ac, a common point is that a long training field (long training field, LTF) sequence that may be used for channel estimation is specified in a physical layer. To improve a system throughput, an orthogonal frequency division multiple access (frequency division multiple access, OFDMA) technology is used in an 802.11ax standard, and an LTF sequence used for channel estimation is also specified in the 802.11ax standard. However, with the development of the mobile Internet and popularization of intelligent terminals, data traffic rapidly increases, and users have increasingly higher requirements for communication service quality. The 802.11ax standard can hardly meet user requirements in aspects such as a high throughput, low jitter, and a low latency. Therefore, there is an urgent need to develop a next-generation WLAN technology, for example, an IEEE 802.11be standard, an extremely high throughput (extremely high throughput, EHT) standard, or a Wi-Fi 7 standard.

~~[0005]~~[0006] For different channel bandwidth (for example, 80 MHz, 160 MHz, 240 MHz, or 320 MHz), how to design an LTF sequence included in a physical layer protocol data unit ((Physical Layer) PHY protocol data unit, PPDU) is an urgent problem to be resolved, so that the LTF sequence has a relatively low peak to average power ratio (peak to average power ratio, PAPR) on entire bandwidth, on a single resource unit, on a combined resource unit, and in a considered multi-stream scenario.

SUMMARY

~~[0006]~~[0007] Embodiments of this application provide a method and an apparatus for transmitting a physical layer protocol data unit, to provide an LTF sequence that has a relatively low PAPR on entire bandwidth, on a single resource unit, on a combined resource unit, and in a considered multi-stream scenario.

~~[0007]~~**[0008]** According to a first aspect, a method for transmitting a physical layer protocol data unit is provided, including: generating a physical layer protocol data unit PPDU, where the PPDU includes a long training field LTF sequence; and sending the PPDU.

~~[0008]~~**[0009]** According to a second aspect, a method for transmitting a physical layer protocol data unit is provided, including: receiving a PPDU; and parsing the received PPDU to obtain a long training field LTF sequence included in the PPDU.

~~[0009]~~**[0010]** With reference to the first aspect and the second aspect, in some implementations, an 80 MHz 2x LTF sequence is:

$2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}, where 2xEHT_LTF_partA is 2xEHT_LTF_partA in DESCRIPTION OF EMBODIMENTS in the specification, 2xEHT_LTF_partB is 2xEHT_LTF_partB in DESCRIPTION OF EMBODIMENTS in the specification, and 0_5 represents 5 consecutive 0s.

~~[0010]~~**[0011]** With reference to the first aspect and the second aspect, in some implementations, a 160 MHz 2x LTF sequence is:

$2xEHT_LTF_160M_{-1012:1012}$ =

{$2xEHT_LTF_80M_{-500:500}$, 0_{23}, $2xEHT_LTF_80M_{-500:500}$};

herein, 0_{23} represents 23 consecutive 0s.

~~[0011]~~**[0012]** With reference to the first aspect and the second aspect, in some implementations, a 240 MHz 2x LTF sequence is:

$2xEHT_LTF_240M_{-1524:1524}$ =

{$-2xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $2xEHT_LTF_80M_{-500:500}$}; or

$2xEHT_LTF_240M_{-1524:1524}$ =

{$-2xEHT_LTF_80M_{-500:500}$, 0_{23}, $2xEHT_LTF_160M_{-1012:1012}$};

herein, $-2xEHT_LTF_80M_{-500:500}$ represents negation of all elements in the sequence $2xEHT_LTF_80M_{-500:500}$;

$-2xEHT_LTF_160M_{-1012:1012}$ represents negation of all elements in the sequence $2xEHT_LTF_160M_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[0012]~~**[0013]** With reference to the first aspect and the second aspect, in some implementations, a 320 MHz 2x LTF sequence is:

$2xEHT_LTF_320M_{-2036:2036}$ =

{$-2xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $2xEHT_LTF_160M_{-1012:1012}$};

herein, $-2xEHT_LTF_160M_{-1012:1012}$ represents negation of all elements in the sequence $2xEHT_LTF_160M_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[0013]~~[0014] With reference to the first aspect and the second aspect, in some implementations, an 80 MHz 2x LTF sequence is a sequence obtained after one or more operations in first operations are performed on $2xEHT_LTF_80M_{-500:500}$;

a 160 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2xEHT_LTF_160M_{-1012:1012}$;

a 240 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2xEHT_LTF_240M_{-1524:1524}$;

a 320 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2xEHT_LTF_320M_{-2036:2036}$; and

the first operations include multiplying elements in a sequence by –1, reversing an order of elements in a sequence, and multiplying an even-numbered or odd-numbered element in non-zero elements in a sequence by –1.

~~[0014]~~[0015] With reference to the first aspect and the second aspect, in some implementations, an 80 MHz 4x LTF sequence is:

$4xEHT_LTF_80M_{-500:500}$ =

{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}, where 4xEHT_LTF_partA is 4xEHT_LTF_partA in DESCRIPTION OF EMBODIMENTS in the specification, 4xEHT_LTF_partB is 4xEHT_LTF_partB in DESCRIPTION OF EMBODIMENTS in the specification, and 0_5 represents 5 consecutive 0s.

~~[0015]~~[0016] With reference to the first aspect and the second aspect, in some implementations, a 160 MHz 4x LTF sequence is:

$4xEHT_LTF_160M_{-1012:1012}$ =

{$-4xEHT_LTF_80M_{-500:500}$, 0_{23}, $4xEHT_LTF_80M_{-500:500}$};

herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[0016]~~[0017] With reference to the first aspect and the second aspect, in some implementations, a 240 MHz 4x LTF sequence is:

$4xEHT_LTF_240M_{-1524:1524}$ =

{$4xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $4xEHT_LTF_80M_{-500:500}$}; or

$4xEHT_LTF_240M_{-1524:1524}$ =

$\{-4xEHT_LTF_80M_{-500:500},\ 0_{23}, 4xEHT_LTF_160M_{-1012:1012}\}$;

herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[0017]~~[0018] With reference to the first aspect and the second aspect, in some implementations, a 320 MHz 4x LTF sequence is:

$4xEHT_LTF_320M_{-2036:2036}$ =

$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23}, -4xEHT_LTF_160M_{-1012:1012}\}$;

herein, 0_{23} represents 23 consecutive 0s, and $-4xEHT_LTF_160M_{-1012:1012}$ represents negation of all elements in the sequence $4xEHT_LTF_160M_{-1012:1012}$.

~~[0018]~~[0019] With reference to the first aspect and the second aspect, in some implementations, an 80 MHz 4x LTF sequence is a sequence obtained after one or more operations in first operations are performed on $4xEHT_LTF_80M_{-500:500}$;

a 160 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $4xEHT_LTF_160M_{-1012:1012}$;

a 240 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $4xEHT_LTF_240M_{-1524:1524}$;

a 320 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $4xEHT_LTF_320M_{-2036:2036}$; and

the first operations include multiplying elements in a sequence by –1, reversing an order of elements in a sequence, and multiplying an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[0019]~~[0020] For the sequence provided in this embodiment of this application, a PAPR in a multi-stream scenario is considered, a PAPR value on a single resource unit (resource unit, RU) is relatively low, a PAPR value on a combined RU is relatively low, and a PAPR value on entire bandwidth is also relatively low.

~~[0020]~~[0021] According to a third aspect, an apparatus for transmitting a physical layer protocol data unit is provided. The apparatus is configured to perform the method provided in any one of the first aspect or the possible implementations of the first aspect. Specifically, the apparatus includes a unit configured to perform any one of the first aspect or the possible implementations of the first aspect.

~~[0021]~~[0022] For example, the apparatus includes:

a processing unit, configured to generate a physical layer protocol data unit PPDU, where the PPDU includes a long training field LTF sequence; and

a transceiver unit, configured to send the PPDU.

~~[0022]~~[0023] According to a fourth aspect, an apparatus for transmitting a physical layer protocol data unit is provided. The apparatus is configured to perform the method provided in any one of the second aspect or the possible implementations of the second aspect. Specifically, the apparatus may include a unit configured to perform any one of the second aspect or the possible implementations of the second aspect.

~~[0023]~~[0024] For example, a transceiver unit is configured to receive a PPDU, and a processing unit is configured to parse the received PPDU to obtain a long training field LTF sequence included in the PPDU.

~~[0024]~~[0025] With reference to the third aspect and the fourth aspect, in some implementations, an 80 MHz 2x LTF sequence is:

$2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}, where 2xEHT_LTF_partA is 2xEHT_LTF_partA in DESCRIPTION OF EMBODIMENTS in the specification, 2xEHT_LTF_partB is 2xEHT_LTF_partB in DESCRIPTION OF EMBODIMENTS in the specification, and 0_5 represents 5 consecutive 0s.

~~[0025]~~[0026] With reference to the third aspect and the fourth aspect, in some implementations, a 160 MHz 2x LTF sequence is:

$2xEHT_LTF_160M_{-1012:1012}$ =

{$2xEHT_LTF_80M_{-500:500}$, 0_{23}, $2xEHT_LTF_80M_{-500:500}$};

herein, 0_{23} represents 23 consecutive 0s.

~~[0026]~~[0027] With reference to the third aspect and the fourth aspect, in some implementations, a 240 MHz 2x LTF sequence is:

$2xEHT_LTF_240M_{-1524:1524}$ =

{$-2xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $2xEHT_LTF_80M_{-500:500}$}; or

$2xEHT_LTF_240M_{-1524:1524}$ =

{$-2xEHT_LTF_80M_{-500:500}$, 0_{23}, $2xEHT_LTF_160M_{-1012:1012}$};

herein, $-2xEHT_LTF_80M_{-500:500}$ represents negation of all elements in the sequence $2xEHT_LTF_80M_{-500:500}$;

$-2\text{xEHT_LTF_160M}_{-1012:1012}$ represents negation of all elements in the sequence $2\text{xEHT_LTF_160M}_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[0027]~~[0028] With reference to the third aspect and the fourth aspect, in some implementations, a 320 MHz 2x LTF sequence is:

$2\text{xEHT_LTF_320M}_{-2036:2036}$ =

$\{-2\text{xEHT_LTF_160M}_{-1012:1012}, 0_{23}, 2\text{xEHT_LTF_160M}_{-1012:1012}\}$;

herein, $-2\text{xEHT_LTF_160M}_{-1012:1012}$ represents negation of all elements in the sequence $2\text{xEHT_LTF_160M}_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[0028]~~[0029] With reference to the third aspect and the fourth aspect, in some implementations, an 80 MHz 2x LTF sequence is a sequence obtained after one or more operations in first operations are performed on $2\text{xEHT_LTF_80M}_{-500:500}$; a 160 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2\text{xEHT_LTF_160M}_{-1012:1012}$; a 240 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2\text{xEHT_LTF_240M}_{-1524:1524}$; a 320 MHz 2x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on $2\text{xEHT_LTF_320M}_{-2036:2036}$; and

the first operations include multiplying elements in a sequence by –1, reversing an order of elements in a sequence, and multiplying an even-numbered or odd-numbered element in non-zero elements in a sequence by –1.

~~[0029]~~[0030] With reference to the third aspect and the fourth aspect, in some implementations, an 80 MHz 4x LTF sequence is:

$4\text{xEHT_LTF_80M}_{-500:500}$ =

$\{4\text{xEHT_LTF_partA}, 0_5, 4\text{xEHT_LTF_partB}\}$, where 4xEHT_LTF_partA is 4xEHT_LTF_partA in DESCRIPTION OF EMBODIMENTS in the specification, 4xEHT_LTF_partB is 4xEHT_LTF_partB in DESCRIPTION OF EMBODIMENTS in the specification, and 0_5 represents 5 consecutive 0s.

~~[0030]~~[0031] With reference to the third aspect and the fourth aspect, in some implementations, a 160 MHz 4x LTF sequence is:

$4\text{xEHT_LTF_160M}_{-1012:1012}$ =

$\{-4\text{xEHT_LTF_80M}_{-500:500}, 0_{23}, 4\text{xEHT_LTF_80M}_{-500:500}\}$;

herein, $-4\text{xEHT_LTF_80M}_{-500:500}$ represents negation of all elements in the sequence

4xEHT_LTF_80M$_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[0031]~~[0032] With reference to the third aspect and the fourth aspect, in some implementations, a 240 MHz 4x LTF sequence is:

4xEHT_LTF_240M$_{-1524:1524}$ =

{4xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, 4xEHT_LTF_80M$_{-500:500}$}; or

4xEHT_LTF_240M$_{-1524:1524}$ =

{–4xEHT_LTF_80M$_{-500:500}$, 0_{23}, 4xEHT_LTF_160M$_{-1012:1012}$};

herein, –4xEHT_LTF_80M$_{-500:500}$ represents negation of all elements in the sequence 4xEHT_LTF_80M$_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[0032]~~[0033] With reference to the third aspect and the fourth aspect, in some implementations, a 320 MHz 4x LTF sequence is:

4xEHT_LTF_320M$_{-2036:2036}$ =

{4xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, –4xEHT_LTF_160M$_{-1012:1012}$};

herein, 0_{23} represents 23 consecutive 0s, and –4xEHT_LTF_160M$_{-1012:1012}$ represents negation of all elements in the sequence 4xEHT_LTF_160M$_{-1012:1012}$.

~~[0033]~~[0034] With reference to the third aspect and the fourth aspect, in some implementations, an 80 MHz 4x LTF sequence is a sequence obtained after one or more operations in first operations are performed on 4xEHT_LTF_80M$_{-500:500}$; a 160 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on 4xEHT_LTF_160M$_{-1012:1012}$; a 240 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on 4xEHT_LTF_240M$_{-1524:1524}$; a 320 MHz 4x LTF sequence is a sequence obtained after one or more operations in the first operations are performed on 4xEHT_LTF_320M$_{-2036:2036}$; and

the first operations include multiplying elements in a sequence by –1, reversing an order of elements in a sequence, and multiplying an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[0034]~~[0035] According to a fifth aspect, an embodiment of this application provides an apparatus for transmitting a physical layer protocol data unit. The apparatus includes a processor and a transceiver that is internally connected to the processor for communication. The processor is configured to generate a physical layer protocol data unit PPDU, where the PPDU includes a long training field LTF sequence. The transceiver is configured to send the PPDU.

~~[0035]~~[0036] The apparatus for transmitting a physical layer protocol data unit provided in the fifth aspect is configured to perform any one of the first aspect or the possible implementations of the first aspect. For specific details, refer to any one of the first aspect or the possible implementations of the first aspect. Details are not described herein.

~~[0036]~~[0037] According to a sixth aspect, an embodiment of this application provides an apparatus for transmitting a physical layer protocol data unit. The apparatus includes a processor and a transceiver that is internally connected to the processor for communication. The transceiver is configured to receive a PPDU. The processor is configured to parse the received PPDU to obtain a long training field LTF sequence included in the PPDU.

~~[0037]~~[0038] The apparatus for transmitting a physical layer protocol data unit provided in the sixth aspect is configured to perform any one of the second aspect or the possible implementations of the second aspect. For specific details, refer to any one of the second aspect or the possible implementations of the second aspect. Details are not described herein.

~~[0038]~~[0039] According to a seventh aspect, an embodiment of this application provides an apparatus for transmitting a physical layer protocol data unit. The apparatus includes a processing circuit and an output interface that is internally connected to the processing circuit for communication. The processing circuit is configured to generate a physical layer protocol data unit PPDU, where the PPDU includes a long training field LTF sequence. The output interface is configured to send the PPDU.

~~[0039]~~[0040] The apparatus for transmitting a physical layer protocol data unit provided in the seventh aspect is configured to perform any one of the first aspect or the possible implementations of the first aspect. For specific details, refer to any one of the first aspect or the possible implementations of the first aspect. Details are not described herein.

~~[0040]~~[0041] According to an eighth aspect, an embodiment of this application provides an apparatus for transmitting a physical layer protocol data unit. The apparatus includes a processing circuit and an input interface that is internally connected to the processing circuit for communication. The input interface is configured to receive a PPDU. The processing circuit is configured to parse the received PPDU to obtain a long training field LTF sequence included in the PPDU.

~~[0041]~~[0042] The apparatus for transmitting a physical layer protocol data unit provided in the eighth aspect is configured to perform any one of the second aspect or the possible

implementations of the second aspect. For specific details, refer to any one of the second aspect or the possible implementations of the second aspect. Details are not described herein.

~~[0042]~~[0043] According to a ninth aspect, an embodiment of this application provides a computer-readable storage medium, configured to store a computer program, where the computer program includes instructions used to perform any one of the first aspect or the possible implementations of the first aspect.

~~[0043]~~[0044] According to a tenth aspect, an embodiment of this application provides a computer-readable storage medium, configured to store a computer program, where the computer program includes instructions used to perform any one of the second aspect or the possible implementations of the second aspect.

~~[0044]~~[0045] According to an eleventh aspect, an embodiment of this application provides a computer program, where the computer program includes instructions used to perform any one of the first aspect or the possible implementations of the first aspect.

~~[0045]~~[0046] According to a twelfth aspect, an embodiment of this application provides a computer program, where the computer program includes instructions used to perform any one of the second aspect or the possible implementations of the second aspect.

BRIEF DESCRIPTION OF DRAWINGS

~~[0046]~~[0047] FIG. 1 is a schematic diagram of a communications system applicable to a method according to an embodiment of this application;

~~[0047]~~[0048] FIG. 2a is a diagram of an internal structure of an access point applicable to an embodiment of this application;

~~[0048]~~[0049] FIG. 2b is a diagram of an internal structure of a station applicable to an embodiment of this application;

~~[0049]~~[0050] FIG. 3 is a schematic diagram of an 80 MHz carrier plan (tone plan) in 802.11ax applicable to an embodiment of this application;

~~[0050]~~[0051] FIG. 4 is a schematic diagram of an 80 MHz carrier plan (tone plan) in 802.11be applicable to an embodiment of this application;

~~[0051]~~[0052] FIG. 5 is a schematic diagram of sequences in 1x, 2x, and 4x modes applicable to an embodiment of this application;

~~[0052]~~**[0053]** FIG. 6 is a schematic diagram of two different types of RU26s applicable to an embodiment of this application;

~~[0053]~~**[0054]** FIG. 7a is a schematic diagram of a division structure and pilot subcarrier distribution of an RU with 20 MHz bandwidth applicable to an embodiment of this application;

~~[0054]~~**[0055]** FIG. 7b is a schematic diagram of a division structure and pilot subcarrier distribution of another RU with 20 MHz bandwidth applicable to an embodiment of this application;

~~[0055]~~**[0056]** FIG. 8 is a schematic diagram of PPDU transmission applicable to an embodiment of this application;

~~[0056]~~**[0057]** FIG. 9 is a schematic diagram of a communications apparatus applicable to an embodiment of this application; and

~~[0057]~~**[0058]** FIG. 10 is a schematic diagram of a communications apparatus applicable to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

~~[0058]~~**[0059]** The following describes in detail the embodiments of this application with reference to the accompanying drawings.

~~[0059]~~**[0060]** To greatly improve a service transmission rate of a WLAN system, the IEEE 802.11ax standard further uses an orthogonal frequency division multiple access (orthogonal frequency division multiple access, OFDMA) technology based on an existing orthogonal frequency division multiplexing (orthogonal frequency division multiplexing, OFDM) technology. The OFDMA technology is a combination of OFDM and FDMA technologies, which is applicable to multi-user access. The OFDM technology is generally applied to a unidirectional broadcast channel, and most actual communications systems support multi-user concurrent communication. Therefore, based on the OFDM technology, the new multiple access technology OFDMA is obtained by allocating one or more subcarrier groups in subcarriers to each user. In OFDMA, a physical channel is divided into a plurality of resource units, each resource unit further includes a plurality of subcarriers (subchannels), and each user may occupy one resource unit for transmission. Therefore, a plurality of users can perform parallel transmission, thereby reducing time overheads and a collision probability of multi-user contention access. In addition, in the

OFDMA technology, because subcarriers overlap each other, spectrum utilization is greatly improved, so that multipath interference and inter-carrier interference can be effectively resisted, and equalization at a receive end is simple. The OFDMA technology supports a plurality of nodes in sending and receiving data simultaneously. This achieves multi-station diversity gains.

~~[0060]~~[0061] In recent years, wireless traffic increases at a very high speed, and users have increasingly higher requirements for communication service quality, such as a low latency and ultra-reliability. As a key technology for carrying a wireless traffic service, a wireless local area network continuously develops and evolves to meet increasingly higher requirements of people for wireless transmission. Existing IEEE 802.11ax can hardly meet user requirements in aspects such as a high throughput, low jitter, and a low latency. Therefore, there is an urgent need to develop a next-generation WLAN technology, for example, an IEEE 802.11be standard, an extremely high throughput (extremely high throughput, EHT) standard, or a Wi-Fi 7 standard, to meet the foregoing extreme performance requirements. The following uses an 802.11be standard as an example for description.

~~[0061]~~[0062] IEEE 802.11be continues to use the OFDMA transmission mode used in 802.11ax. Different from 802.11ax, 802.11ax uses a maximum of 160 MHz bandwidth, but 802.11be uses ultra-high bandwidth of 240 MHz and 320 MHz to achieve an ultra-high transmission rate and support an ultra-dense user scenario.

~~[0062]~~[0063] As everyone knows, OFDM uses a frequency domain equalization technology. Therefore, accuracy of channel estimation greatly affects communication performance. However, an OFDM system has a disadvantage of a high PAPR, and especially in high bandwidth, more subcarriers lead to a more severe PAPR. The high PAPR leads to nonlinear signal distortion and degrades system performance. Because the OFDMA technology is evolved from the OFDM technology, the OFDMA technology inevitably inherits a high-PAPR characteristic of the OFDM technology. Therefore, in an OFDMA system, a PAPR is still an important consideration in a design of an LTF sequence. In particular, because the OFDMA system uses a channel binding technology, in the LTF sequence design, not only a PAPR of the entire sequence is considered, but also a PAPR of the sequence on a single resource unit (resource unit, RU), a PAPR of the sequence on a combined RU, and a PAPR in a considered case of phase rotation caused by a P-matrix when there are a plurality of streams are considered.

~~[0063]~~[0064] In the existing IEEE 802.11ax standard, an LTF that has a low-PAPR

characteristic and that is applicable to resource unit distribution (tone plan) in the 802.11ax standard is designed. Resource unit distribution (tone plan) and pilot positions in the 802.11be standard are different from resource unit distribution (tone plan) and pilot positions in the 802.11ax standard. If an 80 MHz LTF sequence in 802.11ax is directly applied to the 802.11be standard, the LTF sequence has relatively high PAPR values on some resource units, and PAPR values on some resource units are already greater than an average PAPR value of a data part. On the other hand, because a combined RU is used in 802.11be, even if a PAPR value on a single RU is relatively low, a PAPR value on a combined RU obtained by combining a plurality of RUs may also be relatively high. It may be understood that combining a plurality of RUs means allocating the plurality of RUs to one STA. A position of each RU includes a data subcarrier position and a pilot subcarrier position of the RU. Therefore, to make channel estimation more accurate, a low-PAPR LTF sequence for channel estimation needs to be redesigned in IEEE 802.11be.

~~[0064]~~[0065] Based on this, the embodiments of this application provide a method for designing an LTF sequence and a method for transmitting a physical layer protocol data unit PPDU. For an LTF sequence in the embodiments of this application, a PAPR in a multi-stream scenario is considered, a PAPR value on a single RU is relatively low, a PAPR value on a combined RU is relatively low, and a PAPR value on entire bandwidth is also relatively low.

~~[0065]~~[0066] For ease of understanding of technical solutions in the embodiments of this application, the following briefly describes a system architecture of a method for transmitting a PPDU provided in the embodiments of this application. It may be understood that system architectures described in the embodiments of this application are intended to more clearly describe the technical solutions in the embodiments of this application, and does not constitute a limitation on the technical solutions provided in the embodiments of this application.

~~[0066]~~[0067] The technical solutions in the embodiments of this application may be applied to various communications systems, such as a wireless local area network (wireless local area network, WLAN) communications system, a global system for mobile communications (global system for mobile communications, GSM) system, a code division multiple access (code division multiple access, CDMA) system, a wideband code division multiple access (wideband code division multiple access, WCDMA) system, a general packet radio service (general packet radio service, GPRS), a long term evolution (long term evolution, LTE) system, an LTE frequency division duplex (frequency division duplex, FDD) system, LTE time division duplex (time division

duplex, TDD), a universal mobile telecommunications system (universal mobile telecommunications system, UMTS), a worldwide interoperability for microwave access (worldwide interoperability for microwave access, WiMAX) communications system, a subsequent 6th generation (6th generation, 6G) system, or new radio (new radio, NR).

~~[0067]~~[0068] The following is used as an example for description. An application scenario of the embodiments of this application and the method in the embodiments of this application are described only by using a wireless local area network (wireless local area network, WLAN) system as an example.

~~[0068]~~[0069] Specifically, the embodiments of this application may be applied to a wireless local area network (WLAN), and the embodiments of this application may be applied to any protocol in the institute of electrical and electronics engineers (institute of electrical and electronics engineers, IEEE) 802.11 series protocols currently used in the WLAN. The WLAN may include one or more basic service sets (basic service sets, BSSs). A network node in the basic service set includes an access point (access point, AP) and a station (station, STA).

~~[0069]~~[0070] For ease of understanding of the embodiments of this application, a communications system shown in FIG. 1 is first used as an example to describe in detail a communications system to which the embodiments of this application are applicable. A scenario system shown in FIG. 1 may be a WLAN system. The WLAN system in FIG. 1 may include one or more APs and one or more STAs. In FIG. 1, one AP and three STAs are used as an example. Wireless communication may be performed between the AP and each of the STAs according to various standards. For example, wireless communication between the AP and the STA may be performed by using a single-user multiple-input multiple-output (single-user multiple-input multiple-output, SU-MIMO) technology or a multi-user multiple-input multiple-output (multi-user multiple-input multiple-output, MU-MIMO) technology.

~~[0070]~~[0071] Optionally, FIG. 1 is merely a schematic diagram. In addition to being applied to a scenario in which an AP communicates with one or more STAs, the method for transmitting a PPDU provided in the embodiments of this application may be applied to a scenario in which an AP communicates with an AP, and is also applicable to a scenario in which a STA communicates with a STA.

~~[0071]~~[0072] The method for transmitting a PPDU in this application may be implemented by a communications device in a wireless communications system, or a chip or a processor in a

communications device. The communications device may be an access point (access point, AP) device or a station (station, STA) device. The communications device may alternatively be a wireless communications device that supports parallel transmission over a plurality of links. For example, the communications device may be referred to as a multi-link device or a multi-band device (multi-band device). Compared with a communications device that supports only single-link transmission, the multi-link device has higher transmission efficiency and a higher throughput.

~~[0072]~~[0073] An access point (AP) is an apparatus having a wireless communication function, supports communication by using a WLAN protocol, has a function of communicating with another device (for example, a station or another access point) in a WLAN network, and certainly may further have a function of communicating with another device. In a WLAN system, an access point may be referred to as an access point station (AP STA). The apparatus having a wireless communication function may be a device of an entire system, or may be a chip, a processing system, or the like installed in the device of the entire system. The device in which the chip or the processing system is installed may implement the method and functions in the embodiments of this application under control of the chip or the processing system. The AP in the embodiments of this application is an apparatus that provides a service for a STA, and may support 802.11 series protocols. For example, the AP may be a communications entity such as a communications server, a router, a switch, or a network bridge. The AP may include macro base stations, micro base stations, relay stations, or the like in various forms. Certainly, the AP may alternatively be chips and processing systems in the devices in various forms, to implement the method and the functions in the embodiments of this application. The AP is also referred to as a wireless access point, a hotspot, a bridge, or the like. The AP may access a server or a wireless network. APs are access points for mobile users to access wired networks, and are mainly deployed in homes, buildings, and campuses, or are deployed outdoors. The AP is equivalent to a bridge that connects a wired network and a wireless network. A main function of the AP is to connect wireless network clients together, and then connect the wireless network to the Ethernet. Specifically, the AP may be a terminal device or a network device with a wireless fidelity (wireless fidelity, Wi-Fi) chip. Optionally, the AP may be a device that supports a plurality of WLAN standards such as 802.11.

~~[0073]~~[0074] FIG. 2a shows a diagram of an internal structure of an AP product. The AP may have a plurality of antennas or may have a single antenna. In FIG. 2a, the AP includes a physical layer (physical layer, PHY) processing circuit and a media access control (media access control,

MAC) processing circuit. The physical layer processing circuit may be configured to process a physical layer signal, and the MAC layer processing circuit may be configured to process a MAC layer signal.

~~[0074]~~[0075] A station (for example, the STA in FIG. 1) is an apparatus having a wireless communication function, supports communication by using a WLAN protocol, and has a capability of communicating with another station or an access point in a WLAN network. In a WLAN system, a station may be referred to as a non-access point station (non-access point station, non-AP STA). For example, the STA is any user communications device that allows a user to communicate with an AP and then communicate with a WLAN. The apparatus having a wireless communication function may be a device of an entire system, or may be a chip, a processing system, or the like installed in the device of the entire system. The device in which the chip or the processing system is installed may implement the method and functions in the embodiments of this application under control of the chip or the processing system. For example, the STA may be user equipment that may be connected to a network, such as a tablet computer, a desktop computer, a laptop computer, a notebook computer, an ultra-mobile personal computer (Ultra-mobile Personal Computer, UMPC), a handheld computer, a netbook, a personal digital assistant (Personal Digital Assistant, PDA), or a mobile phone, or an Internet of Things node in the Internet of Things, or a vehicle-mounted communications apparatus, entertainment device, game device or system, global positioning system device, or the like in the Internet of Vehicles. The STA may alternatively be a chip and a processing system in the foregoing terminals. A station may also be referred to as a system, a subscriber unit, an access terminal, a mobile station, a remote station, a remote terminal, a mobile device, a user terminal, a terminal, a wireless communications device, a user agent, a user apparatus, or user equipment (user equipment, UE). The STA may be a cellular phone, a cordless phone, a session initiation protocol (session initiation protocol, SIP) phone, a wireless local loop (wireless local loop, WLL) station, a personal digital assistant (personal digital assistant, PDA), a handheld device having a wireless local area network (for example, Wi-Fi) communication function, a wearable device, a computing device, or another processing device connected to a wireless modem.

~~[0075]~~[0076] FIG. 2b shows a structural diagram of a STA with a single antenna. In an actual scenario, the STA may alternatively be a device with a plurality of antennas, and may be a device with more than two antennas. In FIG. 3, the STA may include a physical layer (physical layer,

PHY) processing circuit and a media access control (media access control, MAC) processing circuit. The physical layer processing circuit may be configured to process a physical layer signal, and the MAC layer processing circuit may be configured to process a MAC layer signal.

~~[0076]~~[0077] The foregoing briefly describes the system architecture of the embodiments of this application. For ease of understanding of the embodiments of this application, the following first briefly describes several nouns or terms used in this application.

~~[0077]~~[0078] (1) Subcarrier

~~[0078]~~[0079] Wireless communications signals have limited bandwidth. Bandwidth may be divided, by using the OFDM technology, into a plurality of frequency components within channel bandwidth at a specific frequency interval. These components are referred to as tones. Subscripts of subcarriers are consecutive integers, where a subcarrier whose subscript is 0 corresponds to a direct current component, a subcarrier whose subscript is a negative number corresponds to a frequency component lower than a direct current frequency, and a subcarrier whose subscript is a positive number corresponds to a frequency component higher than the direct current frequency.

~~[0079]~~[0080] (2) 802.11ax carrier plan/resource unit distribution (tone plan)

~~[0080]~~[0081] Resource unit distribution may also be understood as distribution of subcarriers that carry data, and different channel bandwidth may correspond to different tone plans. When OFDMA and multi-user multiple-input multiple-output (multi-user multiple input multiple output, MU-MIMO) technologies are applied, an AP divides spectrum bandwidth into several resource units (resource units, RUs). As specified in the IEEE 802.11ax protocol, spectrum bandwidth of 20 MHz, 40 MHz, 80 MHz, and 160 MHz is divided into a plurality of types of resource units. FIG. 3 is a schematic diagram of an 80 MHz carrier plan (tone plan) in 802.11ax according to an embodiment of this application, including 36 resource unit (resource unit, RU) 26s, or 16 RU52s, or eight RU106s, or four RU242s, or two RU484s, or one RU996 and five direct current subcarriers. There is no gap between a first RU242 and a second RU242. There are seven direct current subcarriers/null subcarriers between the second RU242 and a third RU242. There is also no gap between the third RU242 and a fourth RU242. It should be noted that RUs that can be supported by different total bandwidth have different types and quantities, but in same bandwidth, hybrid-type resource units can be supported.

~~[0081]~~[0082] (3) 802.11be carrier plan/resource unit distribution (tone plan)

~~[0082]~~[0083] In 802.11be, bandwidth is expanded from 160 MHz to 240 MHz and 320 MHz,

to meet requirements of a user for ultra-high bandwidth, an ultra-high transmission rate, and an extremely high throughput. 240 MHz may be considered as direct splicing of three 80 MHz subcarriers in 802.11be, and 320 MHz may be considered as direct splicing of four 80 MHz subcarriers in 802.11be.

~~[0083]~~[0084] FIG. 4 is a schematic diagram of an 80 MHz carrier plan (tone plan) in 802.11be according to an embodiment of this application. 80 MHz bandwidth in 802.11be includes 36 RU26s, or 16 RU52s, or eight RU106s, or four RU242s, or two RU484s and five direct current subcarriers/null subcarriers (that is, two RU489s, where each RU489 includes one RU484 and five direct current subcarriers/null subcarriers), or one RU996 and five direct current subcarriers. There are five direct current subcarriers between a first RU242 and a second RU242. There are also five direct current subcarriers between a third RU242 and a fourth RU242.

~~[0084]~~[0085] It may be understood that the RU26 may refer to a resource unit including 26 subcarriers. It may be further understood that the 26 subcarriers may be consecutive or inconsecutive. Similarly, the RU52 may refer to a resource unit including 52 subcarriers, the RU106 may refer to a resource unit including 106 subcarriers, the RU242 may refer to a resource unit including 242 subcarriers, and so on.

~~[0085]~~[0086] Pilot distribution of the tone plan shown in FIG. 3 and pilot distribution of the tone plan shown in FIG. 4 are also different. Table 1 to Table 6 subsequently describe the pilot distribution of the tone plan shown in FIG. 4. For the pilot distribution of the tone plan shown in FIG. 3, refer to a conventional technology, and details are not described.

~~[0086]~~[0087] In an OFDMA system, a multi-user data packet includes a combination of RUs of a plurality of sizes, and one RU may be allocated to each user. There are the following optional RUs that may be allocated to the user:

(1) an RU including 26 consecutive subcarriers, including: 24 data subcarriers and 2 pilot subcarriers;

(2) an RU including 52 consecutive subcarriers, including: 48 data subcarriers and 4 pilot subcarriers;

(3) an RU including 106 consecutive subcarriers, including: 102 data subcarriers and 4 pilot subcarriers;

(4) an RU including 242 consecutive subcarriers, including: 234 data subcarriers and 8 pilot subcarriers;

(5) an RU including 484 consecutive subcarriers, including: 468 data subcarriers and 16 pilot subcarriers; and

(6) an RU including 996 consecutive subcarriers, including: 980 data subcarriers and 16 pilot subcarriers.

~~[0087]~~[0088] An RU484 is used in 40 MHz multi-user transmission, and a RU996 is used in 80 MHz or 160 MHz multi-user transmission. It should be understood that a 160 MHz tone plan may be considered as two 80 MHz tone plans, a 240 MHz tone plan may be considered as three 80 MHz tone plans, and a 320 MHz tone plan may be considered as four 80 MHz tone plans. Details are not described herein.

~~[0088]~~[0089] The following separately describes positions of different RUs in 80 MHz bandwidth in 802.11be.

~~[0089]~~[0090] (a) In an 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of RU26s are shown in Table 1 below. One RU26 includes 24 data subcarriers and 2 pilot subcarriers.

Table 1: Data subcarrier and pilot subcarrier indexes of RU26s

	Positions of 1st to 18th RU26s	Positions of 19th to 36th RU26s	Pilot Positions
26-tone RU	[–499 –474] [–473 –448] [–445 –420] [–419 –394] [–392 –367] [–365 –340] [–339 –314] [–311 –286] [–285 –260] [–252 –227] [–226 –201] [–198 –173] [–172 –147]	[13 38] [39 64] [67 92] [93 118] [120 145] [147 172] [173 198] [201 226] [227 252] [260 285] [286 311] [314 339] [340 365]	{–494, –480}, {–468, –454}, {–440, –426}, {–414, –400}, {–386, –372}, {–360, –346}, {–334, –320}, {–306, –292}, {–280, –266}, {–246, –232}, {–220, –206}, {–192, –178}, {–166, –152}, {–140, –126}, {–112, –98}, {–86, –72}, {–58, –44}, {–32, –18}, {18, 32}, {44, 58}, {72, 86}, {98, 112}, {126, 140}, {152, 166}, {178, 192}, {206, 220}, {232, 246}, 5DC,

	Positions of 1st to 18th RU26s	Positions of 19th to 36th RU26s	Pilot Positions
	[–145 –120] [–118 –93] [–92 –67] [–64 –39] [–38 –13]	[367 392] [394 419] [420 445] [448 473] [474 499]	{266, 280}, {292, 306}, {320, 334}, {346, 360}, {372, 386}, {400, 414}, {426, 440}, {454, 468}, {480, 494}

~~[0090]~~[0091] Each row in a second column and a third column in Table 1 above indicates one RU26. For example, the last row in the second column indicates an 18th RU26 [–38, –13], and a position of the 18th RU26 is from a subcarrier numbered –38 to a subcarrier numbered –13. For another example, a fifth row in the third column indicates a 23rd RU26 [120, 145], and a position of the 23rd RU26 is from a subcarrier numbered 120 to a subcarrier numbered 145. A fourth column in Table 1 above indicates pilot subcarrier indexes of corresponding 26-tone RUs in sequence. For example, a first 26-tone RU is a subcarrier numbered –499 to a subcarrier numbered –474, and pilot subcarriers are a subcarrier numbered –494 and a subcarrier numbered –480. For another example, a second 26-tone RU is a subcarrier numbered –473 to a subcarrier numbered –448, and pilot subcarriers are a subcarrier numbered –468 and a subcarrier numbered –454. For still another example, a 36th 26-tone RU is a subcarrier numbered 474 to a subcarrier numbered 499, and pilot subcarriers are a subcarrier numbered 480 and a subcarrier numbered 494. It may be understood that the 26-tone RU and the RU26 may be interchangeably used.

~~[0091]~~[0092] (b) In the 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of RU52s are shown in Table 2 below. One RU52 includes 48 data subcarriers and four pilot subcarriers.

Table 2: Data subcarrier and pilot subcarrier indexes of RU52s

	Positions of 1st to 16th RU52s	Pilot Positions
52-tone RU	[–499 –448] [–445 –394] [–365 –314] [–311 –260]	{–494, –480, –468, –454}, {–440, –426, –414, –400}, {–360, –346, –334, –320}, {–306, –292, –280, –266}, {–246, –232, –220, –206},

	Positions of 1st to 16th RU52s	Pilot Positions
	[–252 –201] [–198 –147] [–118 –67] [–64 –13] [13 64] [67 118] [147 198] [201 252] [260 311] [314 365] [394 445] [448 499]	{–192, –178, –166, –152}, {–112, –98, –86, –72}, {–58, –44, –32, –18}, {18, 32, 44, 58}, {72, 86, 98, 112}, {152, 166, 178, 192}, {206, 220, 232, 246}, {266, 280, 292, 306}, {320, 334, 346, 360}, {400, 414, 426, 440}, {454, 468, 480, 494}

~~[0092]~~[0093] Each row in a second column in Table 2 above indicates one RU. For example, a first row in the second column indicates a first RU52 [–499, –448], and a position of the first RU52 is from a subcarrier numbered –499 to a subcarrier numbered –448. A third column in Table 2 above indicates pilot subcarrier indexes of corresponding 52-tone RUs in sequence. For example, a second 52-tone RU is a subcarrier numbered –445 to a subcarrier numbered –394, and pilot subcarriers are a subcarrier numbered –440, a subcarrier numbered –426, a subcarrier numbered –414, and a subcarrier numbered –400. It may be understood that the 52-tone RU and the RU52 may be interchangeably used.

~~[0093]~~[0094] It should be understood that the following tables express the same meaning, and the meaning is not repeated below.

~~[0094]~~[0095] (c) In the 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of RU106s are shown in Table 3 below. One RU106 includes 102 data subcarriers and four pilot subcarriers. It may be understood that a 106-tone RU and the RU106 may be interchangeably used.

Table 3: Data subcarrier and pilot subcarrier indexes of RU106s

	Positions of 1st to 8th RU106s	Pilot Positions
106-tone	[–499 –394]	{–494, –468, –426, –400},

	Positions of 1st to 8th RU106s	Pilot Positions
RU	[–365 –260] [–252 –147] [–118 –13] [13 118] [147 252] [260 365] [394 499]	{–360, –334, –292, –266}, {–246, –220, –178, –152}, {–112, –86, –44, –18}, {18, 44, 86, 112}, {152, 178, 220, 246}, {266, 292, 334, 360}, {400, 426, 468, 494}

~~[0095]~~[0096] (d) In the 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of RU242s are shown in Table 4 below. One RU242 includes 234 data subcarriers and eight pilot subcarriers. It may be understood that a 242-tone RU and the RU242 may be interchangeably used.

Table 4: Data subcarrier and pilot subcarrier indexes of RU242s

	Positions of 1st to 4th RU242s	Pilot Positions
242-tone RU	[–500 –259] [–253 –12] [12 253] [259 500]	{–494, –468, –426, –400, –360, –334, –292, –266}, {–246, –220, –178, –152, –112, –86, –44, –18}, {18, 44, 86, 112, 152, 178, 220, 246}, {266, 292, 334, 360, 400, 426, 468, 494}

~~[0096]~~[0097] (e) In the 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of RU484s are shown in Table 5 below. A 484-tone RU and the RU484 may be interchangeably used. It may be understood that an 80 MHz 484-tone RU in 802.11ax is an RU including 484 consecutive subcarriers; an 80 MHz 484-tone RU in 802.11be is still 468 data subcarriers and 16 pilot subcarriers, but has five direct current subcarriers or null subcarriers in the middle. For example, a first 484-tone RU is subcarriers numbered from –500 to –12, where five direct current subcarriers are numbered –258, –257, –256, –255, and –254, and 16 pilot subcarriers are numbered –494, –468, –426, –400, –360, –334, –292, –266, –246, –220, –178, –152, –112, –86, –44, and –18.

Table 5: Data subcarrier and pilot subcarrier indexes of RU484s

	Positions of 1st to 2nd RU484s	Pilot Positions
484-tone RU	[–500 –259 –253 –12] [12 253 259 500]	{–494, –468, –426, –400, 360, –334, –292, –266, –246, –220, –178, –152, –112, –86, –44, –18}, {18, 44, 86, 112, 152, 178, 220, 246, 266, 292, 334, 360, 400, 426, 468, 494}

~~[0097]~~[0098] (f) In the 80 MHz subcarrier design in FIG. 4, data subcarrier and pilot subcarrier indexes of an RU996 are shown in Table 6 below. A 996-tone RU and the RU996 may be interchangeably used. An 80 MHz 996-tone RU in 802.11be has 980 data subcarriers and 16 pilot subcarriers, and has five direct current subcarriers in the middle. For example, a first 996-tone RU is subcarriers numbered from –500 to 500, where five direct current subcarriers are numbered –2, –1, 0, 1, and 2, and 16 pilot subcarriers are numbered –468, –400, –334, –266, –220, –152, –86, –18, +18, +86, +152, +220, +266, +334, +400, +468.

Table 6: Data subcarrier and pilot subcarrier indexes of an RU996

	Position of the RU996	Pilot Positions
996-tone RU	[–500 –3 3 500]	{–468, –400, –334, –266, –220, –152, –86, –18, +18, +86, +152, +220, +266, +334, +400, +468}

~~[0098]~~[0099] Optionally, an LTF sequence included in the PPDU provided in the embodiments of this application is used in 240 MHz bandwidth and 320 MHz bandwidth, and the 240 MHz bandwidth and the 320 MHz bandwidth may be constructed by using the 80 MHz tone plan shown in FIG. 4. A subcarrier design of 160 MHz bandwidth is based on two 80 MHz tone plans, that is, [RU subcarrier index, pilot position subcarrier index] in 80 MHz – 512: [RU subcarrier index, pilot position subcarrier index] in 80 MHz + 512. Similarly, a subcarrier design of the 240 MHz bandwidth is based on three 80 MHz tone plans. A subcarrier design of the 320 MHz bandwidth is based on two 160 MHz tone plans, that is, [Pilot indices in 160 MHz/pilot indexes in 160 MHz] – 1024: [Pilot indices in 160 MHz/pilot indexes in 160 MHz] + 1024.

~~[0099]~~[00100] (4) Peak to average power ratio

~~[00100]~~**[00101]** Based on observation in time domain, an amplitude of a radio signal continuously varies. Therefore, transmit power of the radio signal is not constant. The peak to average power ratio (peak to average power ratio, PAPR) is a peak to average ratio for short. The peak to average power ratio may be a ratio of an instantaneous power peak value of a continuous signal to an average signal power value in one symbol. The ratio may be represented by using the following formula:

$$\mathrm{PAPR} = 10 \cdot \log_{10}\left(\frac{\max(X_i^2)}{mean(X_i^2)}\right).$$

~~[00101]~~**[00102]** Herein, X_i represents a time domain discrete value of a sequence, $\max(X_i^2)$ represents a maximum value of a square of the time domain discrete value, and $\mathrm{mean}(X_i^2)$ represents an average value of the square of the time domain discrete value.

~~[00102]~~**[00103]** An OFDM symbol is obtained by superimposing a plurality of independently modulated subcarrier signals. When phases of subcarriers are the same or similar, superimposed signals are modulated by a same initial phase signal, resulting in a relatively large instantaneous power peak value. This leads to a relatively large PAPR. An OFDM system has a disadvantage of a high PAPR, and especially in high bandwidth, more subcarriers lead to a more severe PAPR. Because a dynamic range of a general power amplifier is limited, a MIMO-OFDM signal with a relatively large peak to average ratio is very likely to enter a nonlinear area of the power amplifier. The high PAPR causes nonlinear signal distortion, resulting in obvious spectrum spreading interference and in-band signal distortion. This reduces system performance. Therefore, when a sequence is designed, a smaller PAPR of the sequence is better.

~~[00103]~~**[00104]** (5) 4x, 2x, and 1x modes of a long training field sequence

~~[00104]~~**[00105]** To further improve system efficiency in different scenarios, an LTF field needs to support 4x, 2x, and 1x modes. FIG. 5 is a schematic diagram of 4x, 2x, and 1x modes applicable to an embodiment of this application. 20 MHz bandwidth is used as an example. When positions of subcarriers are marked as –128, –127, ..., –2, –1, 0, 1, 2, ..., and 127, subcarriers that are of a 4x HE-LTF element and that carry a long training field sequence are located at –122, –121, ..., –3, –2, 2, 3, ..., 121, and 122. Remaining subcarriers are null subcarriers, and a subcarrier spacing is $\Delta_F^{4\times} = 20MHz/256 = 78.125kHz$. Subcarriers that are of a 2x HE-LTF element and that carry a long training field sequence are located at –122, –120, ..., –4, –2, 2, 4, ..., 120, and 122, and remaining subcarriers are null subcarriers. Equivalently, the positions of the subcarriers may be marked as –

64, –63, ..., –2, –1, 0, 1, 2, ..., and 63. In this case, subcarriers that are of a 2x HE-LTF element and that carry a long training field sequence are located at –61, –60, ..., –2, –1, 1, 2, ..., 60, and 61. Remaining subcarriers are null subcarriers, and a subcarrier spacing is $\Delta_F^{2\times} = 20MHz/128 = 156.25kHz$. Similarly, subcarriers that are of a 1x HE-LTF element and that carry a long training field sequence are centrally located at –120, –116, ..., –8, –4, 4, 8, ..., 116, and 120, and remaining subcarriers are null subcarriers. Equivalently, the positions of the subcarriers may be marked as –32, –31, ..., –2, –1, 0, 1, 2, ..., and 31. In this case, subcarriers that are of a 1x HE-LTF element and that carry a long training field sequence are located at –30, –29, ..., –2, –1, 1, 2, ..., 29, and 30. Remaining subcarriers are null subcarriers, and a subcarrier spacing is $\Delta_F^{1\times} = 20MHz/64 = 312.5kHz$.

~~[00105]~~[00106] That is, four adjacent elements in a sequence form a group. If only one element in the group is not 0, the 1x mode is used. If two elements in the group are not 0, the 2x mode is used. If none of the four elements in the group is 0, the 4x mode is used.

~~[00106]~~[00107] (6) When a Wi-Fi signal is sent in a single-stream pilot mode, a pilot subcarrier and a data subcarrier on each LTF symbol of an LTF field corresponding to the Wi-Fi signal are multiplied by different values, thereby changing a structure of an original LTF sequence. This may cause a PAPR value of the signal of the LTF field to be high when multiplied by some coefficients.

~~[00107]~~[00108] In an OFDM technology, a plurality of LTF fields are used to help a station estimate channels of a plurality of spatial streams (spatial stream). To accurately estimate the channels of the spatial streams and keep LTFs of the streams orthogonal, a Wi-Fi standard proposes to multiply the LTFs by elements of a P-matrix. Specifically, a data subcarrier of an n^{th} LTF symbol sent in an m^{th} spatial stream is multiplied by an element in an m^{th} row and an n^{th} column of the P-matrix, and a pilot subcarrier is multiplied by an element in an m^{th} row and an n^{th} column of an R-matrix. Each row of the R-matrix is equal to a first row of the P-matrix. When the data subcarrier and the pilot subcarrier are multiplied by a same value, a PAPR of an obtained new sequence does not change. When the data subcarrier and the pilot subcarrier are multiplied by different values, a PAPR of an obtained new sequence may change.

~~[00108]~~[00109] A size of the P-matrix is generally 2 × 2, 4 × 4, 6 × 6, 8 × 8, 10 × 10, 12 × 12, 14 × 14, 16 × 16, or the like. For example, when four LTFs need to be sent in one spatial stream, orthogonality may be implemented by using a P-matrix of a size 4 × 4.

~~[00109]~~**[00110]** For example, the P-matrix mainly includes the following types:

$$P_{4x4}=\begin{bmatrix}1 & -1 & 1 & 1\\ 1 & 1 & -1 & 1\\ 1 & 1 & 1 & -1\\ -1 & 1 & 1 & 1\end{bmatrix}$$

$$P_{8x8}=\begin{bmatrix}P_{4x4} & P_{4x4}\\ P_{4x4} & -P_{4x4}\end{bmatrix}$$

$$P_{6\times 6}=\begin{bmatrix}w^{0*0} & -w^{0*1} & w^{0*2} & w^{0*3} & w^{0*4} & -w^{0*5}\\ w^{1*0} & -w^{1*1} & w^{1*2} & w^{1*3} & w^{1*4} & -w^{1*5}\\ w^{2*0} & -w^{2*1} & w^{2*2} & w^{2*3} & w^{2*4} & -w^{2*5}\\ w^{3*0} & -w^{3*1} & w^{3*2} & w^{3*3} & w^{3*4} & -w^{3*5}\\ w^{4*0} & -w^{4*1} & w^{4*2} & w^{4*3} & w^{4*4} & -w^{4*5}\\ w^{5*0} & -w^{5*1} & w^{5*2} & w^{5*3} & w^{5*4} & -w^{5*5}\end{bmatrix} \quad w=\exp(-j2\pi/6)$$

~~[00110]~~**[00111]** Elements in P-matrices of different sizes are different, and may indicate different rotated phases. For example, elements in P-matrices of sizes 4 * 4, 8 * 8, and 16 * 16 are all 1 and –1, and correspond to a same rotated phase. For example, pilot position * 1, and non-pilot position * 1; or pilot position * 1, and non-pilot position * –1; or pilot position * –1, and non-pilot position * 1; or pilot position * –1, and non-pilot position * –1. When the pilot position and the non-pilot position are multiplied by a same value, a PAPR that is of a sequence obtained after phase rotation and that is on a single RU, on a combined RU, or on entire bandwidth does not change relative to a PAPR of an original sequence. When the pilot position and the non-pilot position are multiplied by different values, a PAPR that is of a sequence obtained after phase rotation and that is on a single RU, on a combined RU, or on entire bandwidth changes relative to a PAPR of an original sequence. Generally, four sequences with different PAPRs may be obtained after phase rotation is performed on one sequence.

~~[00111]~~**[00112]** In this application, phase rotation is considered for the LTF sequence, and a PAPR of an obtained rotated sequence is relatively low on a single RU, on a combined RU, and on entire bandwidth. Therefore, a PAPR of the sequence is relatively low in a multi-stream scenario.

~~[00112]~~**[00113]** (7) PAPR value of an HE-LTF sequence with 80 MHz bandwidth in 802.11ax applied to a resource unit with 80 MHz bandwidth in 802.11be

~~[00113]~~**[00114]** It may be learned from comparison between FIG. 3 and FIG. 4 that there is no spacing between a first RU242 and a second RU242 in 802.11ax, and there are five direct current subcarriers between a first RU242 and a second RU242 in 802.11be. Therefore, if the HE-LTF

sequence with 80 MHz bandwidth in 802.11ax is directly applied to 80 MHz bandwidth in 802.11be, PAPR values on some resource units in 802.11be are relatively high. As shown in Table 7 below, a second row in Table 7 represents average PAPR values of a data part on different resource units, a third row represents PAPR values on different resource units when a 2x HE-LTF sequence with 80 MHz bandwidth in 802.11ax is applied to 80 MHz bandwidth in 802.11be, and a fourth row represents PAPR values on different resource units when a 4x HE-LTF sequence with 80 MHz bandwidth in 802.11ax is applied to 80 MHz bandwidth in 802.11be. It may be learned from Table 7 that PAPR values on resource units (RU26, RU52, and RU484 + RU242) in a second column, a third column, and the last column in Table 7 are greater than average PAPR values of the data part.

Table 7: PAPR value of an LTF field on each resource unit when an 80 MHz LTF in 802.11ax is applied to an 802.11be standard shown in FIG. 4

Max PAPR	RU26	RU52	RU106	RU242	RU484	RU996	RU52 + RU26	RU106 + RU26	RU484 + RU242
Data (16QAM)	6.52	7.17	7.76	8.36	8.82	9.22	7.54	7.94	9.08
LTF 2x	7.95	7.43	6.71	8.23	7.33	6.48	6.37	6.64	9.65
LTF 4x	7.29	8.48	6.69	7.07	7.36	6.74	6.90	7.42	8.46

~~[00114]~~[00115] (8) Type A and Type B RUs

~~[00115]~~[00116] By analyzing resource unit division and pilot positions in 802.11be in 80 MHz bandwidth in Table 1 (data subcarrier and pilot subcarrier indexes of RU26s), it may be found that, in the 36 RU26s, pilot subcarriers of some RU26 resource units are located at 6^{th} and 20^{th} subcarriers of the 26 subcarriers, as shown in FIG. 6. This type of RU26 is referred to as a Type A RU26 in this application. However, pilot subcarriers of some other RU26 resource units are located at 7^{th} and 21^{st} subcarriers of the 26 subcarriers. This type of RU26 is referred to as a Type B RU26 in this application.

~~[00116]~~[00117] For example, a first 26-tone RU is a subcarrier numbered –499 to a subcarrier

numbered –474, and pilot subcarriers are a subcarrier numbered –494 and a subcarrier numbered –480. For another example, a second 26-tone RU is a subcarrier numbered –473 to a subcarrier numbered –448, and pilot subcarriers are a subcarrier numbered –468 and a subcarrier numbered –454. That is, pilot subcarriers are located at 6^{th} and 20^{th} subcarriers of the 26 subcarriers. For still another example, a fifth 26-tone RU is a subcarrier numbered –392 to a subcarrier numbered –367, and pilot subcarriers are a subcarrier numbered –386 and a subcarrier numbered –372. That is, pilot subcarriers are located at 7^{th} and 21^{st} subcarriers of the 26 subcarriers.

~~[00117]~~[00118] Correspondingly, an RU52 including two RU26s also has two types. One type is an RU52 including two Type A RU26s, and this type of RU52 is referred to as a Type A RU52 in this application. The other type is an RU52 including two Type B RU26s, and this type of RU52 is referred to as a Type B RU52 in this application.

~~[00118]~~[00119] Correspondingly, an RU106 including two RU52s also has two types. One type is an RU106 including two Type A RU52s, and this type of RU106 is referred to as a Type A RU106 in this application. The other type is an RU106 including two Type B RU52s, and this type of RU106 is referred to as a Type B RU106 in this application.

~~[00119]~~[00120] Correspondingly, an RU242 including nine RU26s has a more complicated structure, but also has two types. As shown in FIG. 7a, one type of RU242 includes eight Type A RU26s and one Type B RU26, and this type of RU242 is referred to as a Type A RU242 in this application. As shown in FIG. 7b, the other type of RU242 includes eight Type B RU26s and one Type A RU26, and this type of RU242 is referred to as a Type B RU242 in this application.

~~[00120]~~[00121] Further, it is found that an only difference between a Type A resource unit and a Type B resource unit lies in that pilot positions are different. If a head and a tail of one type of resource unit are reversed (which may also be referred to as reversed sorting), the other type of resource unit is obtained. That is, a reverse order of a Type A resource unit is a Type B resource unit, and a reverse order of a Type B resource unit is a Type A resource unit. For example, pilot subcarriers of a Type A RU26 are located at 6^{th} and 20^{th} subcarriers of the 26 subcarriers, and pilot subcarriers of a Type B RU26 are located at 7^{th} and 21^{st} subcarriers of the 26 subcarriers. If a Type A RU26 is viewed in a reverse order, pilot subcarriers are located at 7^{th} and 21^{st} subcarriers of the 26 subcarriers, that is, a Type B RU26 is viewed. If a Type B RU26 is viewed in a reverse order, pilot subcarriers are located at 6^{th} and 20^{th} subcarriers of the 26 subcarriers, that is, a Type A RU26 is viewed. An example of order reversing or head-tail reversing is further described for ease of

understanding. For example, if Type A is 1, 2, 3, and 4, Type B is 4, 3, 2, and 1.

~~[00121]~~[00122] The foregoing describes content related to the embodiments of this application. The following describes, in detail with reference to more accompanying drawings, the method for transmitting a PPDU provided in the embodiments of this application. The embodiments of this application may be applied to a plurality of different scenarios, including the scenario shown in FIG. 1, but are not limited to the scenario. For example, for uplink transmission, a STA may be used as a transmit end, and an AP may be used as a receive end. For downlink transmission, the AP may be used as a transmit end, and the STA may be used as a receive end. For other transmission scenarios, for example, for data transmission between APs, one AP may be used as a transmit end, and the other AP may be used as a receive end. For another example, for data transmission between STAs, one STA may be used as a transmit end, and the other STA may be used as a receive end. In the embodiments of this application, the method is described by using a first communications device and a second communications device. It may be understood that the first communications device may be an AP or a STA (for example, the AP or the STA shown in FIG. 1), and the second communications device may also be an AP or a STA (for example, the AP or the STA shown in FIG. 1).

~~[00122]~~[00123] The embodiments of this application provide a plurality of possible LTF sequences. These LTF sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. It may be understood that a smaller PAPR value indicates a lower requirement on a linear power amplifier and better performance.

Embodiment 1

~~[00123]~~[00124] Embodiment 1 of this application describes a possible procedure of the method for transmitting a physical layer protocol data unit PPDU provided in this application.

~~[00124]~~[00125] Referring to FIG. 8, FIG. 8 is a schematic flowchart of a method 800 for transmitting a PPDU according to an embodiment of this application. The method 800 shown in FIG. 8 may include but is not limited to the following steps:

~~[00125]~~[00126] S810: A first communications device generates a physical layer protocol data unit PPDU, where the PPDU includes a long training field LTF, and the long training field carries an LTF sequence.

~~[00126]~~[00127] Specifically, a method for generating the LTF sequence by the first communications device is subsequently described.

~~[00127]~~[00128] S820: The first communications device sends the PPDU. Correspondingly, a second communications device receives the PPDU.

~~[00128]~~[00129] S830: The second communications device parses the PPDU to obtain the LTF sequence in the PPDU. For a specific parsing manner, refer to an existing description. No limitations are imposed herein.

~~[00129]~~[00130] It may be understood that the "LTF sequence" mentioned in this application may be a frequency domain sequence of an LTF, or may be referred to as a frequency domain sequence of a long training field.

~~[00130]~~[00131] Then, a method for generating the LTF sequence in S810 is described. Specifically, the following steps are included.

~~[00131]~~[00132] Step 1: Determine a Type A RU26 basic sequence set S_{RU26}^{A} and a Type B RU26 basic sequence set S_{RU26}^{B}.

~~[00132]~~[00133] A1. A sequence of an appropriate length is selected based on an application scenario and an application requirement of the LTF sequence. The length is generally a sequence length corresponding to a minimum resource unit in a tone plan. For example, when the minimum resource unit is an RU26, a length of a basic sequence is 26 bits.

~~[00133]~~[00134] Generally, an element of an LTF sequence is limited to 1 or –1, and certainly, is not limited to this. Only a case of two elements 1 or –1 is considered for an EHT LTF sequence. In a case of 1x, there are 2^6 = 64 possible sequences that are of a Type A RU26 and that may be selected. In a case of 2x, there are 2^13 = 8192 possible sequences that are of a Type A RU26 and that may be selected. In a case of 4x, there are 2^26 = 8192 * 8192 possible sequences that are of a Type A RU26 and that may be selected. A sequence of a Type B RU26 resource unit has a same quantity of possibilities.

~~[00134]~~[00135] For a possible sequence of a Type A RU26, all possible rotated sequences of the sequence of the Type A RU26 are determined (PAPRs of all possible rotated sequences in this application include a PAPR of an original sequence). Then, a PAPR value of each rotated sequence

on the Type A RU26 may be calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of the Type A RU26), the sequence of the Type A RU26 may be used as a basic sequence that meets a condition. The basic sequence may be added to the Type A RU26 basic sequence set.

~~[00135]~~[00136] The foregoing process is repeated, to traverse each possible sequence of the Type A RU26 and add a basic sequence with a relatively low PAPR value to the Type A RU26 basic sequence set. Optionally, a quantity of basic sequences in the Type A RU26 basic sequence set may be limited. If a relatively large quantity of basic sequences are selected by specifying a threshold, some sequences whose PAPR values are relatively low may be selected from sequences whose PAPRs are less than the specified threshold and added to the Type A RU26 basic sequence set.

~~[00136]~~[00137] PAPR values of sequences in the Type A RU26 basic sequence set are relatively low on the Type A RU26, and PAPR values of rotated sequences of these sequences are also relatively low on the Type A RU26.

~~[00137]~~[00138] B1. A sequence set formed after head-tail reversing (reversed sorting) is performed on all basic sequences in the Type A RU26 basic sequence set is the Type B RU26 basic sequence set S_{RU26}^{B}.

~~[00138]~~[00139] There are two basic principles: One basic principle is that a PAPR value of a new sequence formed after head-tail reversing of any sequence is the same as a PAPR value of the original sequence. The other basic principle is that, after head-tail reversing is performed on either of a sequence corresponding to a Type A resource unit and a sequence corresponding to a Type B resource unit in this application, a position of a pilot subcarrier of the sequence exactly corresponds to a pilot position of the sequence corresponding to the resource unit of the other type. A PAPR of a sequence is not changed after head-tail reversing is performed on the sequence, and a pilot point of a Type A sequence after head-tail reversing exactly corresponds to a position of a pilot point of a Type B sequence. Therefore, the sequence set formed after head-tail reversing is performed on all basic sequences in the Type A RU26 basic sequence set is the Type B RU26 basic sequence set. PAPRs of sequences in the Type B RU26 basic sequence set are relatively low on the Type B RU26, and PAPR values of rotated sequences of these basic sequences are also relatively low on the Type B RU26.

~~[00139]~~**[00140]** Step 2: Determine a Type A RU52 basic sequence set S_{RU52}^{A} and a Type B RU52 basic sequence set S_{RU52}^{B}.

~~[00140]~~**[00141]** A2. Two RU26 basic sequences are selected from the Type A RU26 basic sequence set obtained in A1 of step 1, and are spliced into a sequence of a Type A RU52. PAPR values that are on the Type A RU52 and that are of a plurality of rotated sequences corresponding to the spliced sequence of the Type A RU52 are calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of the Type A RU52), the sequence of the Type A RU52 may be used as a basic sequence that meets a condition. The basic sequence may be added to the Type A RU52 basic sequence set.

~~[00141]~~**[00142]** The foregoing process is repeated, to traverse each possible sequence of the Type A RU52 and add a basic sequence with a relatively low PAPR value to the Type A RU52 basic sequence set. Optionally, a quantity of sequences in the Type A RU52 basic sequence set may be limited. If a relatively large quantity of basic sequences are selected by specifying a threshold, some sequences whose PAPR values are relatively low may be selected from sequences whose PAPRs are less than the specified threshold and added to the Type A RU52 basic sequence set.

~~[00142]~~**[00143]** PAPR values of sequences in the Type A RU52 basic sequence set obtained in this manner are relatively low on the Type A RU52, and PAPR values of rotated sequences of these basic sequences are also relatively low on the Type A RU52. In addition, PAPR values on all sub-RU resource units (for example, Type A RU26s) included in the Type A RU52 are also relatively low.

~~[00143]~~**[00144]** B2. A sequence set formed after head-tail reversing (reversed sorting) is performed on all basic sequences in the Type A RU52 basic sequence set is the Type B RU52 basic sequence set S_{RU52}^{B}. PAPR values of basic sequences in the Type B RU52 basic sequence set are relatively low on the Type B RU52, and PAPR values of rotated sequences of these basic sequences are also relatively low on the Type B RU52. In addition, PAPR values on all sub-RU resource units (for example, Type B RU26s) included in the Type B RU52 are also relatively low.

~~[00144]~~**[00145]** Step 3: Determine a Type A RU106 basic sequence set S_{RU106}^{A} and a Type B RU106 basic sequence set S_{RU106}^{B}.

~~[00145]~~**[00146]** A3. Two RU52 basic sequences are selected from the Type A RU52 basic sequence set obtained in A2 of step 2, and are spliced into a sequence of a Type A RU106 by

considering possible values corresponding to two intermediate idle carriers. PAPR values that are on the Type A RU106 and that are of a plurality of rotated sequences corresponding to the spliced sequence of the Type A RU106 are calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of the Type A RU106), the sequence of the Type A RU106 may be used as a basic sequence that meets a condition. The basic sequence may be added to the Type A RU106 basic sequence set.

~~[00146]~~[00147] The foregoing process is repeated, to traverse each possible sequence of the Type A RU106 and add a basic sequence with a relatively low PAPR value to the Type A RU106 basic sequence set. Optionally, a quantity of sequences in the Type A RU106 basic sequence set may be limited. A limiting manner is the same as the limiting manner of the Type A RU26 basic sequence set described above, and details are not described again.

~~[00147]~~[00148] PAPR values of basic sequences in the Type A RU106 basic sequence set obtained in this manner are relatively low on the Type A RU106, and PAPR values of rotated sequences of these basic sequences are also relatively low on the Type A RU106. In addition, PAPR values on all sub-RU resource units (for example, Type A RU52s and Type A RU26s) included in the Type A RU106 are also relatively low.

~~[00148]~~[00149] B3. A sequence set formed after head-tail reversing (reversed sorting) is performed on all basic sequences in the Type A RU106 basic sequence set is the Type B RU106 basic sequence set S_{RU106}^{B}. PAPR values of basic sequences in the Type B RU106 basic sequence set are relatively low on the Type B RU106, and PAPR values of rotated sequences of these basic sequences are also relatively low on the Type B RU106. In addition, PAPR values on all sub-RU resource units (for example, Type B RU52s and Type B RU26s) included in the Type B RU106 are also relatively low.

~~[00149]~~[00150] Step 4: Determine a basic sequence set S_{MRU106}^{A} and a basic sequence set S_{MRU106}^{B}.

~~[00150]~~[00151] A4. An RU106 basic sequence is selected from the Type A RU106 basic sequence set obtained in A3 of step 3, and a basic sequence of a first Type A RU52 in the RU106 basic sequence and a basic sequence of a Type A RU26 (a first Type A RU26 in a second Type A RU52) adjacent to the first Type A RU52 are spliced into a multiple-resource unit (MRU) sequence. PAPR values that are on the corresponding Type A RU52 + RU26 and that are of a plurality of

rotated sequences corresponding to the spliced multiple-resource unit sequence are calculated. If the PAPR value of each rotated sequence is less than a specified threshold, the multiple-resource unit sequence may be used as a basic sequence that meets a condition. The multiple-resource unit sequence may be added to the basic sequence set S^{A}_{MRU106}.

~~[00151]~~[00152] Each possible multiple-resource unit sequence is traversed, and the foregoing process is performed to add a basic sequence with a relatively low PAPR value to the basic sequence set S^{A}_{MRU106}. Optionally, a quantity of sequences in the basic sequence set S^{A}_{MRU106} may be limited. A limiting manner is the same as the limiting manner of the Type A RU26 basic sequence set described above, and details are not described again.

~~[00152]~~[00153] Each sequence in the basic sequence set S^{A}_{MRU106} and a corresponding rotated sequence have relatively low PAPRs on a corresponding Type A RU106, and have relatively low PAPRs on all sub-RU resource units (for example, Type A RU52s and Type A RU26s) included in the Type A RU106. In addition, the multiple-resource unit sequence obtained by combining the first Type A RU52 and the Type A RU26 adjacent to the first Type A RU52 also has a relatively low PAPR value on the corresponding Type A RU52 + RU26.

~~[00153]~~[00154] B4. A sequence set formed after head-tail reversing (reversed sorting) is performed on all basic sequences in the basic sequence set S^{A}_{MRU106} is S^{B}_{MRU106}.

~~[00154]~~[00155] Each sequence in the basic sequence set S^{B}_{MRU106} and a corresponding rotated sequence have relatively low PAPRs on a corresponding Type B RU106, and have relatively low PAPRs on all sub-RU resource units (for example, Type B RU52s and Type B RU26s) included in the Type B RU106. In addition, a multiple-resource unit sequence obtained by combining a first Type B RU52 and a Type B RU26 adjacent to the first Type B RU52 also has a relatively low PAPR value on a corresponding Type B RU.

~~[00155]~~[00156] Step 5: Determine a basic sequence set $S_{RU106A+RU26B}$ and a basic sequence set $S_{RU26A+RU106B}$.

~~[00156]~~[00157] A5. An EHT standard supports a resource unit combination of a Type A RU106 and a Type B RU26. Therefore, one basic sequence may be separately selected from S^{A}_{RU106} obtained in A3 of step 3 and S^{B}_{RU26} obtained in B1 of step 1 for splicing. A PAPR value of each rotated sequence corresponding to a spliced sequence is calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of a combined RU of the Type A RU106 and the

Type B RU26), the spliced sequence may be used as a basic sequence that meets a condition, and may be added to $S_{RU106A+RU26B}$.

~~[00157]~~[00158] The foregoing process is repeated, to traverse each possible spliced sequence and add a basic sequence with a relatively low PAPR value to the basic sequence set $S_{RU106A+RU26B}$. Optionally, a quantity of sequences in the basic sequence set $S_{RU106A+RU26B}$ may be limited. A limiting manner is the same as the limiting manner of the Type A RU26 basic sequence set described above, and details are not described again.

~~[00158]~~[00159] Each sequence in the basic sequence set $S_{RU106A+RU26B}$ and a corresponding rotated sequence have relatively low PAPRs on the corresponding Type A RU106, have relatively low PAPRs on all sub-RU resource units (for example, Type A RU52s and Type A RU26s) included in the Type A RU106, have relatively low PAPRs on the corresponding Type B RU26, and also have relatively low PAPRs on the combined RU of the Type A RU106 and the Type B RU26.

~~[00159]~~[00160] B5. The EHT standard also supports a resource unit combination of a Type A RU26 and a Type B RU106. Therefore, a sequence set formed after head-tail reversing (reversed sorting) is performed on all sequences in the basic sequence set $S_{RU106A+RU26B}$ is $S_{RU26A+RU106B}$. Each sequence in the basic sequence set $S_{RU26A+RU106B}$ and a corresponding rotated sequence have relatively low PAPRs on the corresponding Type B RU106, have relatively low PAPRs on all sub-RU resource units (for example, Type B RU52s and Type B RU26s) included in the Type B RU106, have relatively low PAPRs on the corresponding Type A RU26, and also have relatively low PAPRs on a combined RU of the Type B RU106 and the Type A RU26.

~~[00160]~~[00161] Step 6: Determine a Type A RU242 basic sequence set S_{RU242}^{A} and a Type B RU242 basic sequence set S_{RU242}^{B}.

~~[00161]~~[00162] A6. One basic sequence is separately selected from the basic sequence set $S_{RU106A+RU26B}$ obtained in A5 of step 5 and the basic sequence sequence set S_{MRU106}^{A} obtained in A4 of step 4, and the selected basic sequences are spliced into a sequence of a Type A RU242 by considering possible values corresponding to four intermediate idle carriers. PAPR values that are on the Type A RU242 and that are of a plurality of rotated sequences corresponding to the spliced sequence of the Type A RU242 are calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of the Type A RU242), the sequence of the Type A RU242 may be used as a basic sequence that meets a condition. The basic sequence may be added to the Type A

RU242 basic sequence set.

~~[00162]~~[00163] The foregoing process is repeated, to traverse each possible sequence of the Type A RU242 and add a basic sequence with a relatively low PAPR value to the Type A RU242 basic sequence set. Optionally, a quantity of sequences in the Type A RU242 basic sequence set may be limited. A limiting manner is the same as the limiting manner of the Type A RU26 basic sequence set described above, and details are not described again.

~~[00163]~~[00164] A basic sequence in the Type A RU242 basic sequence set obtained in this manner and a corresponding rotated sequence have relatively low PAPR values on the Type A RU242, and have a low-PAPR characteristic of the basic sequence set $S_{RU106A+RU26B}$ described in A5 of step 5 and a low-PAPR characteristic of S^{A}_{MRU106} described in A4 of step 4. Specifically, the basic sequence in the Type A RU242 basic sequence set and the rotated sequence have relatively low PAPR values on all sub-RU resource units (for example, Type A RU52s, Type A RU26s, and Type A RU106s) included in the Type A RU242, and also have relatively low PAPR values on all combined RUs included in the Type A RU242 (for example, a combination of a Type A RU106 and a Type B RU26, and a combination of a first Type A RU52 in a Type A RU106 and an adjacent Type A RU26).

~~[00164]~~[00165] B6. A sequence set formed after head-tail reversing (reversed sorting) is performed on all basic sequences in the Type A RU242 basic sequence set is the Type B RU242 basic sequence set S^{B}_{RU242}.

~~[00165]~~[00166] A basic sequence in the Type B RU242 basic sequence set obtained in this manner and a rotated sequence have relatively low PAPR values on the Type B RU242, and have a low-PAPR characteristic of the basic sequence set $S_{RU106\mathrm{B}+RU26\mathrm{A}}$ described in B5 of step 5 and a low-PAPR characteristic of S^{B}_{MRU106} described in B4 of step 4. For specific content, refer to the foregoing description, and details are not described again.

~~[00166]~~[00167] Step 7: Determine an RU484 basic sequence set S_{RU484}.

~~[00167]~~[00168] One basic sequence is separately selected from the basic sequence set S^{A}_{RU242} obtained in A6 of step 6 and the basic sequence set S^{B}_{RU242} obtained in B6 of step 6, and the selected basic sequences are spliced into a sequence of an RU484. PAPR values that are on the RU484 and that are of a plurality of rotated sequences corresponding to the sequence of the RU484 are calculated. If the PAPR value of each rotated sequence is less than a specified threshold (the specified threshold may be a PAPR average value, median value, or the like of a data part of the

Type A RU242), the sequence of the RU484 may be used as a basic sequence that meets a condition. The basic sequence may be added to the RU484 basic sequence set S_{RU484}.

~~[00168]~~[00169] The foregoing process is repeated, to traverse each possible sequence of the RU484 and add a basic sequence with a relatively low PAPR value to the RU484 basic sequence set S_{RU484}. Optionally, a quantity of sequences in the RU484 basic sequence set may be limited. A limiting manner is the same as the limiting manner of the Type A RU26 basic sequence set described above, and details are not described again.

~~[00169]~~[00170] A sequence in the RU484 basic sequence set obtained in this manner and a corresponding rotated sequence have relatively low PAPR values on the RU484, and have a low-PAPR characteristic of the basic sequence set S_{RU242}^{A} described in A6 of step 6 and a low-PAPR characteristic of S_{RU242}^{B} described in B6 of step 6. For specific content, refer to the foregoing description, and details are not described again.

~~[00170]~~[00171] Step 8: Determine an RU996 basic sequence set S_{RU996}.

~~[00171]~~[00172] Two basic sequences are randomly selected from the basic sequence set S_{RU484} obtained in step 7, and are spliced into a sequence of an RU996. PAPR values that are on the RU996 and that are of a plurality of rotated sequences corresponding to the sequence of the RU996 are calculated, and PAPRs that are on a combined RU and that are of the plurality of rotated sequences corresponding to the sequence of the RU996 are calculated. The combined RU includes a combination of a first RU484 in the RU996 and a first RU242 in a second RU484, a combination of the first RU484 in the RU996 and a second RU242 in the second RU484, a combination of the second RU484 in the RU996 and a first RU242 in the first RU484, and a combination of the second RU484 in the RU996 and a second RU242 in the first RU484.

~~[00172]~~[00173] If the PAPR values are all less than a specified threshold corresponding to each RU, the spliced sequence of the RU996 is used as a basic sequence that meets a condition. The basic sequence may be added to the RU996 basic sequence set. The specified threshold corresponding to each RU may be a PAPR average value, median value, or the like of a data part of each RU. For example, a threshold corresponding to a PAPR value of a rotated sequence on the RU996 is a PAPR average value, median value, or the like of a data part of the RU996. For another example, a threshold corresponding to a PAPR value of a rotated sequence on a combined RU of the first RU484 in the RU996 and the first RU242 in the second RU484 is a PAPR average value, median value, or the like of a data part of the combined RU of the first RU484 in the RU996 and

the first RU242 in the second RU484.

~~[00173]~~[00174] It should be noted that the obtained sequence of the RU996 herein does not include a sequence value on a subcarrier between any two RU242s, and the sequence value on the subcarrier between any two RU242s only affects a PAPR of the sequence corresponding to the complete RU996, without affecting a PAPR of another smaller RU or a combined RU.

~~[00174]~~[00175] The foregoing process is repeated, to traverse each possible sequence of the RU996 and add a basic sequence with a relatively low PAPR value to the RU996 basic sequence set S_{RU996}. A sequence in the RU996 basic sequence set obtained in this manner and a corresponding rotated sequence have relatively low PAPR values on the RU996, have relatively low PAPR values on a combined RU of any RU242 and any RU484 in the RU996, and have a low-PAPR characteristic of the basic sequence set S_{RU484} obtained in step 7.

~~[00175]~~[00176] Short basic sequences are gradually spliced, and a combination with a relatively low PAPR value is selected, to obtain a longer sequence with a low-PAPR characteristic. PAPRs of sequences within different RU sizes and sequences of multi-RU combinations are considered in the splicing from short to long. Therefore, when a formed sequence is used as an LTF, the sequence has a relatively low PAPR for a plurality of different resource unit sizes and a combination of a plurality of resource units. In a sequence selection process, symmetry of different RU types is used, thereby greatly reducing search computation of sequences.

~~[00176]~~[00177] After the foregoing process, LTF sequences with 80 MHz bandwidth may be obtained. The LTF sequences with 80 MHz bandwidth have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

~~[00177]~~[00178] In addition, further combination splicing may be performed to obtain an LTF sequence with bandwidth greater than 80 MHz (for example, 160 MHz bandwidth, 240 MHz bandwidth, or 320 MHz bandwidth). Certainly, further combination splicing may alternatively not be performed, but sequences with 80 MHz bandwidth are directly spliced. For example, an LTF sequence with 160 MHz bandwidth may be directly formed by splicing two sequences with 80 MHz bandwidth, an LTF sequence with 240 MHz bandwidth may be directly formed by splicing

three sequences with 80 MHz bandwidth, and an LTF sequence with 320 MHz bandwidth may be directly formed by splicing four sequences with 80 MHz bandwidth. The LTF sequences with 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth obtained in this manner also meet a characteristic that the sequences and corresponding rotated sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

~~[00178]~~[00179] Then, LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in 2x and 4x modes are described.

~~[00179]~~[00180] (1) A possible 2x LTF sequence with 80 MHz bandwidth is denoted by 2xEHT_LTF_80M. Subcarrier numbers of the sequence 2xEHT_LTF_80M range from –500 to 500.

~~[00180]~~[00181] For example, $2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB}, where –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 2xEHT_LTF_partA includes 498 elements, and 2xEHT_LTF_partB includes 498 elements.

~~[00181]~~[00182] For example, the sequence 2xEHT_LTF_partB is obtained by reversing an order of the sequence 2xEHT_LTF_partA and then negating an even-numbered element in non-zero elements, that is, negating an element at a position that is an integer multiple of 4.

~~[00182]~~[00183] In an example, 2xEHT_LTF_partA =

{–1 0 –1 0 –1 0 1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 –1 0 1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
1 0 –1 0 1 0 –1 0 –1 0 1 0 1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 –1 0 1 0 –1 0 –1 0 –1 0
1 0 1 0 1 0 –1 0 –1 0 –1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
1 0 1 0 –1 0 –1 0 –1 0 –1 0 1 0
–1 0 1 0 1 0 –1 0 1 0 1 0 1 0
1 0 –1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 –1 0 1 0 1 0 –1 0 1 0

–1 0 1 0 1 0 –1 0 1 0 –1 0 –1 0

–1 0 –1 0 –1 0 –1 0 –1 0 1 0 –1 0

–1 0 –1 0 1 0 1 0 1 0 1 0 –1 0

1 0 1 0 –1 0 1 0 –1 0 1 0 1 0

1 0 –1 0 1 0 –1 0 –1 0 –1 0 1 0

1 0 –1 0 1 0 –1 0 –1 0 –1 0 1 0

–1 0 –1 0 1 0 –1 0 –1 0 1 0 1 0

1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0

–1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0

–1 0 –1 0 –1 0 –1 0 1 0 1 0 –1 0

–1 0 –1 0 1 0 1 0 1 0 1 0 1 0

1 0 –1 0 1 0 –1 0 –1 0 1 0 –1 0

1 0 –1 0 1 0 –1 0 –1 0 1 0 1 0

–1 0 –1 0 –1 0 –1 0 1 0 1 0 –1 0

–1 0 –1 0 –1 0 –1 0 1 0 –1 0 1 0

–1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1 0

–1 0 –1 0 1 0 –1 0 1 0 1 0 –1 0

–1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0

–1 0 –1 0 1 0 –1 0 –1 0 –1 0 1 0

1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0

1 0 1 0 –1 0 –1 0 1 0 –1 0 1 0

–1 0 –1 0 –1 0 1 0 1 0 1 0 –1 0

1 0 1 0 1 0 1 0 1 0 –1 0 1 0

1 0 1 0 –1 0 –1 0}

2xEHT_LTF_partB =

{0 –1 0 1 0 1 0 –1 0 1 0 1

0 1 0 –1 0 1 0 –1 0 1 0 1 0 1

0 –1 0 1 0 1 0 –1 0 1 0 1 0 1

0 1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 –1 0 1 0 1 0 –1 0 –1 0 –1 0 1

0 1 0 1 0 1 0 1 0 –1 0 1 0 1

0 1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 1 0 1 0 1 0 1 0 –1 0 1 0 –1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 –1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1

0 –1 0 1 0 1 0 –1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 –1 0 –1 0 –1 0 –1 0 1 0 1

0 –1 0 1 0 1 0 1 0 1 0 –1 0 1

0 1 0 1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 1 0 –1 0 1 0 1 0 –1 0 1

0 –1 0 –1 0 –1 0 1 0 1 0 1 0 1

0 –1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 –1 0 1 0 1 0 –1 0 –1 0 1 0 1

0 –1 0 1 0 1 0 1 0 –1 0 1 0 –1

0 1 0 1 0 1 0 –1 0 –1 0 –1 0 1

0 1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 –1 0 1 0 –1 0 1 0 –1 0 –1 0 1

0 1 0 –1 0 –1 0 1 0 –1}

~~[00183]~~[00184] It may be learned from 2xEHT_LTF_partA and 2xEHT_LTF_partB that:

1. 2xEHT_LTF_partA is (only the last 20 elements are listed):

... 1, 0, 1, 0, 1, 0, 1, 0, –1, 0, 1, 0, 1, 0, 1, 0, –1, 0, –1, 0.

2. A reverse order of 2xEHT_LTF_partA is:

0, –1, 0, –1, 0, 1, 0, 1, 0, 1, 0, –1, 0, 1, 0, 1, 0, 1, 0, 1 ....

3. 2xEHT_LTF_partB is obtained by negating an even-numbered element in non-zero elements in the reverse order of 2xEHT_LTF_partA, and 2xEHT_LTF_partB is 0, –1, 0, 1, 0, 1, 0, –1, 0, 1, 0, 1, 0, 1, 0, –1, 0, 1, 0, –1, ...

~~[00184]~~[00185] That is, the sequence 2xEHT_LTF_partB is obtained by reversing an order of the sequence 2xEHT_LTF_partA and then negating an element at a position that is in the sequence and that is an integer multiple of 4.

~~[00185]~~[00186] (2) A possible 2x LTF sequence with 160 MHz bandwidth is denoted by 2xEHT_LTF_160M. 2xEHT_LTF_160M may be constructed based on 2xEHT_LTF_80M described in (1), and subcarrier numbers of the sequence 2xEHT_LTF_160M range from –1012 to 1012.

~~[00186]~~[00187] For example, $2xEHT_LTF_160M_{-1012:1012}$ =

$$\{2xEHT_LTF_80M_{-500:500},\ 0_{23},\ 2xEHT_LTF_80M_{-500:500}\}.$$

~~[00187]~~[00188] Herein, $2xEHT_LTF_80M_{-500:500}$ in $2xEHT_LTF_160M_{-1012:1012}$ is $2xEHT_LTF_80M_{-500:500}$ in (1); and 0_{23} represents 23 consecutive 0s.

~~[00188]~~[00189] (3) A possible 2x LTF sequence with 320 MHz bandwidth is denoted by 2xEHT_LTF_320M. 2xEHT_LTF_320M is constructed based on 2xEHT_LTF_160M described in (2), and subcarrier numbers of the sequence 2xEHT_LTF_320M range from –2036 to 2036.

~~[00189]~~[00190] For example, $2xEHT_LTF_320M_{-2036:2036}$ =

$$\{-2xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ 2xEHT_LTF_160M_{-1012:1012}\}.$$

~~[00190]~~[00191] Herein, $-2xEHT_LTF_160M_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence $2xEHT_LTF_160M_{-1012:1012}$; $2xEHT_LTF_160M_{-1012:1012}$ in $2xEHT_LTF_320M_{-2036:2036}$ is $2xEHT_LTF_160M_{-1012:1012}$ in (2); and 0_{23} represents 23 consecutive 0s.

~~[00191]~~[00192] (4) A possible 2x LTF sequence with 240 MHz bandwidth is denoted by 2xEHT_LTF_240M. 2xEHT_LTF_240M is constructed based on 2xEHT_LTF_160M described in (2) and 2xEHT_LTF_80M described in (1). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be

continuous or discontinuous in frequency domain. A punctured 2x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 2xEHT_LTF_240M range from –1524 to 1524.

~~[00192]~~[00193] For example, $2xEHT_LTF_240M_{-1524:1524}$ =

$$\{-2xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ 2xEHT_LTF_80M_{-500:500}\}.$$

~~[00193]~~[00194] For example, $2xEHT_LTF_240M_{-1524:1524}$ =

$$\{-2xEHT_LTF_80M_{-500:500},\ 0_{23},\ 2xEHT_LTF_160M_{-1012:1012}\}.$$

~~[00194]~~[00195] Herein, $-2xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $2xEHT_LTF_80M_{-500:500}$; $-2xEHT_LTF_160M_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence $2xEHT_LTF_160M_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[00195]~~[00196] In $2xEHT_LTF_240M_{-1524:1524}$ in the two examples in (4), $2xEHT_LTF_80M_{-500:500}$ is $2xEHT_LTF_80M_{-500:500}$ in (1), and $2xEHT_LTF_160M_{-1012:1012}$ is $2xEHT_LTF_160M_{-1012:1012}$ in (2).

~~[00196]~~[00197] (5) Sequences obtained by performing one or more of the following operations on the 2x LTF sequences with various bandwidth described in (1) to (4), for example, $2xEHT_LTF_80M_{-500:500}$, $2xEHT_LTF_160M_{-1012:1012}$, $2xEHT_LTF_240M_{-1524:1524}$, and $2xEHT_LTF_320M_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00197]~~[00198] Operation (1): Multiply elements in a sequence by –1.

~~[00198]~~[00199] Operation (2): Reverse an order of elements in a sequence. It is assumed that a sequence is 123456, and a sequence obtained by reversing an order of elements in the sequence is 654321.

~~[00199]~~[00200] Operation (3): Multiply an even-numbered or odd-numbered element in non-zero elements by –1.

~~[00200]~~[00201] As shown in Table 8 below, Table 8 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 2xEHT_LTF_320M described in (3) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00201]~~**[00202]** For example, an RU26 is used as an example. An 80 MHz sequence includes 36 RU26s, that is, 36 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 36 * 4 = 144 PAPRs may be obtained by considering rotated sequences. A maximum value of the 144 PAPRs is 5.26. 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M are obtained based on the sequence 2xEHT_LTF_80M. PAPR values of 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M on the RU26s are the same as the PAPR values of 2xEHT_LTF_80M on the RU26s.

~~[00202]~~**[00203]** It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00203]~~**[00204]** It may be learned from Table 8 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 8: Comparison between PAPR maximum values of 2x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU26	5.26	5.89
RU52	4.58	6.71
RU52 + RU26	5.19	7.10

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU106	5.19	7.29
RU106 + RU26	5.98	7.44
RU242	5.42	7.94
RU484	5.51	8.44
RU484 + RU242	6.98	8.83
RU996	5.78	8.84
RU996 + RU484	7.52	9.17
RU2*996	8.54 or 8.56	9.27 or 9.28
RU2*996 + RU484	8.64 or 9.15	9.50
RU3*996	9.04	9.54
RU3*996 + RU484	9.48	9.55
RU4*996	9.07	9.60

~~[00204]~~[00205] The LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode are described above. Then, LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode are described.

~~[00205]~~[00206] (6) A possible 4x LTF sequence with 80 MHz bandwidth is denoted by 4xEHT_LTF_80M. Subcarrier numbers of the sequence 4xEHT_LTF_80M range from –500 to 500.

~~[00206]~~[00207] For example, $4xEHT_LTF_80M_{-500:500}$ = {4xEHT_LTF_partA, 0_5 , 4xEHT_LTF_partB}.

~~[00207]~~[00208] Herein, –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 4xEHT_LTF_partA includes 498 elements, and 4xEHT_LTF_partB includes 498 elements.

~~[00208]~~[00209] For example, the sequence 4xEHT_LTF_partB is obtained by reversing an order of the sequence 4xEHT_LTF_partA and then negating an element value at an even-numbered position, that is, negating an element at a position that is an integer multiple of 2.

~~[00209]~~[00210] In an example, 4xEHT_LTF_partA =

{–1 –1 –1 1 1 –1 1 1 –1 –1 1 –1 1 –1

–1 1 –1 1 –1 1 1 1 1 1 –1 –1 –1 –1

–1 1 1 –1 –1 1 –1 –1 1 –1 1 –1 –1 1

–1 1 –1 1 –1 1 1 1 –1 –1 –1 –1 –1 –1

1 1 –1 –1 –1 –1 1 1 –1 –1 –1 1 –1 1

1 1 –1 –1 1 –1 –1 –1 –1 1 –1 1 –1 –1

1 1 –1 1 –1 –1 1 1 1 –1 1 –1 –1 –1

1 1 1 1 1 1 1 –1 1 –1 1 1 1 –1

–1 –1 1 –1 –1 –1 –1 1 1 –1 –1 1 –1 –1

1 –1 –1 1 –1 1 –1 1 1 –1 1 –1 1 –1

1 –1 –1 –1 –1 1 1 –1 1 1 –1 1 1 –1

1 1 1 1 –1 –1 –1 1 –1 1 1 1 1 1

1 1 –1 1 –1 –1 –1 –1 1 –1 1 1 1 –1

–1 –1 1 1 –1 –1 1 1 1 1 –1 1 1 –1

–1 –1 1 1 1 –1 1 –1 1 1 –1 –1 –1 –1

–1 –1 1 –1 –1 –1 –1 –1 1 1 –1 1 1 –1

1 1 –1 –1 –1 1 1 –1 –1 –1 –1 –1 –1 1

–1 1 –1 1 1 1 –1 1 1 –1 –1 1 –1 1

1 –1 –1 1 –1 –1 –1 1 1 1 –1 1 –1 1

1 –1 1 –1 –1 –1 1 1 –1 1 –1 1 1 –1

–1 1 1 –1 1 1 1 –1 1 –1 1 1 1 1

–1 –1 –1 –1 –1 –1 –1 1 1 –1 –1 1 1 –1

1 1 1 1 –1 –1 –1 –1 1 –1 –1 –1 –1 –1

–1 1 1 –1 1 –1 –1 –1 –1 –1 –1 –1 –1 –1

1 1 1 –1 –1 –1 1 1 –1 1 1 –1 –1 1

–1 1 –1 1 1 1 –1 1 1 1 –1 1 –1 –1

1 –1 1 –1 –1 –1 –1 –1 –1 1 –1 –1 1 1

–1 –1 –1 1 1 1 1 1 –1 –1 –1 1 1 1

1 –1 1 1 –1 –1 –1 1 –1 –1 1 1 –1 1

–1 1 1 1 1 1 –1 –1 1 1 1 1 1 –1

1 1 –1 –1 –1 1 –1 –1 –1 1 –1 1 –1 1

1 1 1 1 1 1 1 1 –1 –1 –1 1 1 –1

1 1 1 –1 –1 1 –1 1 1 –1 –1 1 –1 1

–1 –1 –1 –1 –1 –1 –1 1 1 –1 –1 1 –1 –1

–1 1 1 –1 1 –1 1 1 1 –1 1 –1 –1 1

–1 –1 1 1 1 –1 1 1}

4xEHT_LTF_partB =

{1 –1 –1 –1 1 –1 –1 1 1 1 –1 –1

–1 –1 1 –1 –1 –1 –1 –1 1 1 –1 1 1 1

–1 –1 1 1 –1 1 –1 1 –1 1 1 1 1 1

–1 –1 1 1 1 1 –1 –1 1 –1 –1 –1 1 1

–1 1 1 –1 1 –1 1 –1 1 –1 1 1 1 1

1 1 –1 1 1 1 –1 1 1 –1 –1 –1 1 –1

1 –1 –1 1 1 –1 1 –1 1 1 1 1 1 –1

–1 1 1 1 –1 1 1 –1 –1 –1 1 –1 1 1

–1 1 1 –1 1 –1 1 1 –1 1 1 –1 –1 1

1 1 –1 1 –1 1 –1 –1 –1 –1 –1 1 1 1

1 –1 1 1 1 –1 1 1 1 1 1 1 –1 –1

1 1 1 –1 –1 1 –1 –1 1 –1 –1 1 –1 1

–1 1 –1 1 –1 –1 –1 –1 1 1 –1 1 –1 1

–1 –1 –1 1 –1 1 1 –1 1 –1 –1 –1 1 1

–1 –1 1 1 –1 1 –1 1 –1 1 1 –1 1 –1

–1 –1 –1 –1 1 –1 –1 –1 1 1 –1 –1 1 1

1 1 1 –1 –1 1 –1 –1 –1 –1 1 1 1 1

1 –1 1 1 1 –1 1 1 1 1 1 1 –1 –1

1 1 1 –1 –1 1 –1 –1 1 –1 –1 1 –1 1

–1 1 –1 1 –1 –1 –1 –1 1 1 –1 1 –1 1

–1 –1 –1 1 –1 1 1 –1 1 –1 –1 –1 1 1

–1 –1 1 1 –1 1 –1 1 –1 1 1 –1 1 –1

1 1 1 –1 –1 1 –1 –1 –1 –1 1 1 1 1

1 –1 1 1 1 –1 1 1 1 1 1 1 –1 1

–1 –1 1 1 1 –1 1 1 1 1 1 1 –1 1

–1 1 –1 –1 1 1 –1 1 1 –1 –1 –1 1 1

1 –1 –1 1 –1 1 –1 –1 –1 1 –1 1 –1 1

1 –1 –1 –1 –1 –1 1 –1 –1 1 –1 –1 1 1

1 –1 1 –1 –1 1 1 –1 –1 1 –1 –1 1 –1

–1 –1 –1 1 –1 1 1 1 1 –1 1 –1 1 –1

1 1 1 1 –1 1 1 –1 1 –1 –1 –1 1 1

1 –1 –1 –1 1 1 –1 1 –1 –1 –1 –1 –1 –1

–1 –1 1 1 1 1 1 1 –1 –1 –1 1 1 1

–1 –1 1 1 –1 1 –1 –1 –1 1 –1 –1 1 –1

–1 –1 –1 –1 1 –1 1 –1 1 –1 –1 1 –1 –1

–1 –1 1 –1 –1 1 1 1 1 –1 –1 1 1 1

1 1 –1 1 –1 –1 –1 1 1 –1 1 1 1 1

–1 1 1 –1 –1 1 –1 1 1 –1 –1 1 –1 1

–1 1 1 –1 1 1 1 1 1 1 1 –1 –1

–1 –1 –1 1 1 1 –1 –1 1 1 –1 1 –1 1

1 –1 1 –1 1 1 1 1 1 1 –1 –1 –1 –1

–1 1 1 –1 –1 –1 1 1 –1 1}

~~[00210]~~[00211] (7) A possible 4x LTF sequence with 160 MHz bandwidth is denoted by 4xEHT_LTF_160M. 4xEHT_LTF_160M may be constructed based on 4xEHT_LTF_80M described in (6), and subcarrier numbers of the sequence 4xEHT_LTF_160M range from –1012 to 1012.

~~[00211]~~[00212] For example, $4xEHT_LTF_160M_{-1012:1012}$ =

$$\{-4xEHT_LTF_80M_{-500:500},\ 0_{23},\ 4xEHT_LTF_80M_{-500:500}\}.$$

~~[00212]~~[00213] Herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation (multiplied by –1) of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; 0_{23} represents 23 consecutive 0s; and $4xEHT_LTF_80M_{-500:500}$ in $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_80M_{-500:500}$ in (6).

~~[00213]~~[00214] (8) A possible 4x LTF sequence with 320 MHz bandwidth is denoted by 4xEHT_LTF_320M. 4xEHT_LTF_320M is constructed based on 4xEHT_LTF_160M described in (7), and subcarrier numbers of the sequence 4xEHT_LTF_320M range from –2036 to 2036.

~~[00214]~~[00215] For example, $4xEHT_LTF_320M_{-2036:2036}$ =

$$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ -4xEHT_LTF_160M_{-1012:1012}\}.$$

~~[00215]~~[00216] Herein, $-4xEHT_LTF_160M_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_160M_{-1012:1012}$; 0_{23} represents 23 consecutive 0s; and $4xEHT_LTF_160M_{-1012:1012}$ in $4xEHT_LTF_320M_{-2036:2036}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (7).

~~[00216]~~[00217] (9) A possible 4x LTF sequence with 240 MHz bandwidth is denoted by 4xEHT_LTF_240M. 4xEHT_LTF_240M is constructed based on 4xEHT_LTF_160M described in (7) and 4xEHT_LTF_80M described in (6). For example, when an 80 MHz channel in a 320

MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 4x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 4xEHT_LTF_240M range from –1524 to 1524.

~~[00217]~~[00218] For example, $4xEHT_LTF_240M_{-1524:1524}$ =

$$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ 4xEHT_LTF_80M_{-500:500}\}.$$

~~[00218]~~[00219] For example, $4xEHT_LTF_240M_{-1524:1524}$ =

$$\{-4xEHT_LTF_80M_{-500:500},\ 0_{23},\ 4xEHT_LTF_160M_{-1012:1012}\}.$$

~~[00219]~~[00220] Herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[00220]~~[00221] In $4xEHT_LTF_240M_{-1524:1524}$ in the two examples in (9), $4xEHT_LTF_80M_{-500:500}$ is $4xEHT_LTF_80M_{-500:500}$ in (6), and $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (7).

~~[00221]~~[00222] (10) Sequences obtained by performing one or more of the following operations on the 4x LTF sequences with various bandwidth described in (6) to (9), for example, $4xEHT_LTF_80M_{-500:500}$, $4xEHT_LTF_160M_{-1012:1012}$, $4xEHT_LTF_240M_{-1524:1524}$, and $4xEHT_LTF_320M_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00222]~~[00223] Operation (1): Multiply elements in a sequence by –1.

~~[00223]~~[00224] Operation (2): Reverse an order of elements in a sequence.

~~[00224]~~[00225] Operation (3): Multiply an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[00225]~~[00226] As shown in Table 9 below, Table 9 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 4xEHT_LTF_320M described in (8) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00226]~~[00227] For example, an RU52 is used as an example. An 80 MHz sequence includes 16 RU52s, that is, 16 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 16 * 4 = 64 PAPRs may be obtained by considering rotated sequences. A maximum value of the 64

PAPRs is 4.97. 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M are obtained based on the sequence 4xEHT_LTF_80M. PAPR values of 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M on the RU52s are the same as the PAPR values of 4xEHT_LTF_80M on the RU52s.

~~[00227]~~[00228] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00228]~~[00229] It may be learned from Table 9 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 9: Comparison between PAPR maximum values of 4x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR	Median PAPR of BPSK Data
RU26	3.98	5.89
RU52	4.97	6.71
RU52 + RU26	5.41	7.10
RU106	5.45	7.29
RU106 + RU26	5.65	7.44
RU242	5.66	7.94
RU484	6.53	8.44

RU Size	Max PAPR	Median PAPR of BPSK Data
RU484 + RU242	7.62	8.83
RU996	6.36	8.84
RU996 + RU484	8.39	9.17
RU2*996	8.71 or 9.37	9.27 or 9.28
RU2*996 + RU484	9.04 or 9.57	9.50
RU3*996	9.37	9.54
RU3*996 + RU484	9.79	9.55
RU4*996	9.40	9.60

~~[00229]~~[00230] (11) A possible 4x LTF sequence with 80 MHz bandwidth is denoted by 4xEHT_LTF_80M. Subcarrier numbers of the sequence 4xEHT_LTF_80M range from –500 to 500.

~~[00230]~~[00231] For example, $4xEHT_LTF_80M_{-500:500}$ = {4xEHT_LTF_partA, 0_5 , 4xEHT_LTF_partB}.

~~[00231]~~[00232] Herein, –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 4xEHT_LTF_partA includes 498 elements, and 4xEHT_LTF_partB includes 498 elements.

~~[00232]~~[00233] For example, the sequence 4xEHT_LTF_partB is obtained by reversing an order of the sequence 4xEHT_LTF_partA and then negating an element value at an even-numbered position, that is, negating an element at a position that is an integer multiple of 2.

~~[00233]~~[00234] In an example, 4xEHT_LTF_partA =

{–1 –1 –1 –1 –1 –1 –1 –1 1 1 1 –1 –1 1
1 –1 1 –1 –1 –1 1 –1 1 1 –1 1 –1 –1
1 –1 1 –1 1 –1 –1 –1 –1 –1 1 1 –1 –1
1 1 –1 –1 1 –1 1 1 1 1 –1 –1 –1 1
–1 –1 1 –1 –1 1 –1 –1 –1 –1 –1 1 1 1
1 1 –1 1 –1 1 –1 –1 1 1 1 –1 1 –1
1 1 1 1 –1 1 –1 1 1 1 1 1 –1 1
–1 –1 1 1 1 –1 –1 1 1 1 –1 –1 –1 –1
1 1 1 1 1 –1 1 1 –1 1 1 1 –1 –1
1 1 –1 1 –1 1 –1 1 –1 –1 1 1 –1 –1

1 –1 1 1 1 –1 –1 –1 –1 –1 –1 –1 –1 1

–1 1 –1 1 1 –1 1 1 1 1 –1 1 1 1

1 –1 –1 –1 1 1 1 1 1 1 –1 1 –1 –1

1 1 –1 1 –1 –1 1 1 1 –1 1 –1 1 1

1 1 –1 1 –1 –1 1 1 –1 1 1 1 1 –1

–1 1 –1 –1 –1 1 –1 1 –1 –1 –1 1 1 1

–1 1 –1 –1 1 1 1 –1 1 1 –1 –1 –1 –1

–1 –1 –1 1 –1 –1 –1 1 –1 1 1 –1 –1 1

–1 1 1 1 1 1 1 –1 –1 1 1 1 1 –1

1 1 1 –1 1 –1 –1 –1 1 1 1 –1 –1 1

1 –1 1 –1 –1 1 1 –1 –1 –1 1 1 1 1

1 1 –1 1 –1 1 1 –1 1 –1 –1 1 –1 1

1 –1 –1 –1 1 1 1 1 1 1 –1 1 –1 1

–1 1 –1 1 1 1 –1 1 1 1 –1 –1 1 –1

–1 –1 1 –1 –1 –1 –1 –1 1 1 –1 1 –1 –1

1 –1 –1 –1 –1 1 1 –1 –1 –1 –1 1 –1 1

1 1 1 –1 1 –1 –1 1 1 –1 –1 –1 –1 –1

1 –1 –1 1 –1 1 1 –1 1 1 1 1 1 –1

1 –1 –1 –1 –1 1 –1 1 –1 –1 –1 1 1 –1

–1 –1 1 –1 –1 1 –1 –1 1 –1 1 1 –1 –1

–1 1 –1 –1 –1 1 –1 –1 –1 –1 –1 1 1 1

1 1 –1 –1 1 –1 –1 –1 –1 1 1 1 1 1

–1 –1 1 1 1 –1 1 –1 –1 1 1 –1 1 1

–1 –1 –1 –1 1 1 1 –1}

4xEHT_LTF_partB =

{–1 –1 1 –1 –1 1 –1 1 1 –1 1 1 1 –1

1 1 1 1 –1 1 1 1 1 1 –1 –1 –1 –1

–1 1 1 –1 –1 1 1 –1 1 –1 –1 –1 1 1

–1 –1 –1 –1 1 –1 –1 1 1 –1 1 –1 1 1

–1 1 –1 –1 –1 1 1 –1 1 –1 1 1 –1 1

–1 1 1 1 –1 1 1 1 –1 1 1 –1 –1 –1

–1 1 1 1 –1 –1 –1 1 –1 –1 1 1 –1 1

1 1 1 1 –1 1 –1 –1 –1 –1 1 –1 1 –1

5 –1 –1 1 1 1 1 –1 –1 –1 1 –1 1 –1 –1

1 1 –1 –1 –1 –1 1 –1 1 1 1 1 –1 1

–1 –1 1 1 –1 1 –1 –1 –1 1 1 1 1 –1

–1 1 –1 1 –1 –1 –1 1 –1 –1 –1 1 1 –1

1 1 1 –1 1 1 1 1 1 1 1 1 –1

10 1 –1 1 –1 –1 1 –1 –1 1 1 1 1 –1 –1

–1 –1 1 1 1 1 1 –1 1 –1 1 –1 –1 1

–1 –1 1 1 –1 –1 –1 –1 1 1 –1 –1 1 –1

–1 1 –1 –1 –1 –1 1 –1 –1 –1 1 –1 1 1

–1 –1 1 –1 1 –1 1 1 1 1 –1 –1 1 1

15 1 1 –1 1 1 1 –1 1 –1 1 –1 1 1 –1

–1 –1 1 –1 –1 1 1 1 1 –1 1 1 –1 1

1 1 1 1 –1 1 1 1 –1 –1 1 –1 1 1

1 –1 –1 1 1 1 1 –1 1 –1 –1 –1 –1 –1

1 –1 –1 1 1 1 1 –1 –1 1 1 1 1 –1

20 1 –1 1 –1 –1 1 –1 –1 1 –1 1 1 1 –1

1 –1 –1 –1 1 1 1 1 1 1 –1 1 –1 1

–1 1 –1 –1 1 –1 –1 –1 –1 1 1 –1 –1 1

1 1 1 1 1 1 1 –1 –1 1 1 –1 1 1

1 –1 –1 –1 1 –1 1 –1 –1 1 –1 1 1 –1

25 1 1 –1 –1 1 –1 –1 1 1 1 1 –1 1 –1

1 1 1 1 1 –1 1 –1 –1 –1 –1 –1 1 –1

–1 1 1 1 1 1 1 –1 1 –1 1 1 –1 1

–1 1 1 1 –1 –1 –1 1 1 1 –1 1 1 –1

1 –1 –1 –1 –1 1 1 –1 –1 1 1 –1 –1 1

30 –1 1 –1 –1 –1 –1 –1 –1 –1 1 1 1 1 –1

–1 –1 –1 1 –1 –1 –1 –1 1 1 –1 –1 1 –1

–1 1 –1 1 –1 1 –1 1}

~~[00234]~~[00235] (12) A possible 4x LTF sequence with 160 MHz bandwidth is denoted by 4xEHT_LTF_160M. 4xEHT_LTF_160M may be constructed based on 4xEHT_LTF_80M described in (11), and subcarrier numbers of the sequence 4xEHT_LTF_160M range from –1012 to 1012.

~~[00235]~~[00236] For example, $4xEHT_LTF_160M_{-1012:1012}$ =

{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, –4xEHT_LTF_partB}.

~~[00236]~~[00237] Herein, –4xEHT_LTF_partB represents negation (multiplied by –1) of all elements in the sequence 4xEHT_LTF_partB; 0_{23} represents 23 consecutive 0s; 0_5 represents five consecutive 0s; and 4xEHT_LTF_partA and 4xEHT_LTF_partB in $4xEHT_LTF_160M_{-1012:1012}$ are 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (11).

~~[00237]~~[00238] (13) A possible 4x LTF sequence with 320 MHz bandwidth is denoted by 4xEHT_LTF_320M. 4xEHT_LTF_320M is constructed based on 4xEHT_LTF_160M described in (12), and subcarrier numbers of the sequence 4xEHT_LTF_320M range from –2036 to 2036.

~~[00238]~~[00239] For example, $4xEHT_LTF_320M_{-2036:2036}$ =

$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ -4xEHT_LTF_160M_{-1012:1012}\}$.

~~[00239]~~[00240] Herein, $-4xEHT_LTF_160M_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_160M_{-1012:1012}$; 0_{23} represents 23 consecutive 0s; and $4xEHT_LTF_160M_{-1012:1012}$ in $4xEHT_LTF_320M_{-2036:2036}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (12).

~~[00240]~~[00241] (14) A possible 4x LTF sequence with 240 MHz bandwidth is denoted by 4xEHT_LTF_240M. 4xEHT_LTF_240M is constructed based on 4xEHT_LTF_160M described in (12) and 4xEHT_LTF_80M described in (11). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 4x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 4xEHT_LTF_240M range from –1524 to 1524.

~~[00241]~~[00242] For example, $4xEHT_LTF_240M_{-1524:1524}$ =

$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ -4xEHT_LTF_80M_{-500:500}\}$.

~~[00242]~~[00243] Herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1)

of all elements in the sequence 4xEHT_LTF_80M$_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[00243]~~[00244] In 4xEHT_LTF_240M$_{-1524:1524}$ in the two examples in (14), 4xEHT_LTF_80M$_{-500:500}$ is 4xEHT_LTF_80M$_{-500:500}$ in (11), and 4xEHT_LTF_160M$_{-1012:1012}$ is 4xEHT_LTF_160M$_{-1012:1012}$ in (12).

~~[00244]~~[00245] (15) Sequences obtained by performing one or more of the following operations on the 4x LTF sequences with various bandwidth described in (11) to (14), for example, 4xEHT_LTF_80M$_{-500:500}$, 4xEHT_LTF_160M$_{-1012:1012}$, 4xEHT_LTF_240M$_{-1524:1524}$, and 4xEHT_LTF_320M$_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00245]~~[00246] Operation (1): Multiply elements in a sequence by –1.

~~[00246]~~[00247] Operation (2): Reverse an order of elements in a sequence.

~~[00247]~~[00248] Operation (3): Multiply an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[00248]~~[00249] As shown in Table 10 below, Table 10 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 4xEHT_LTF_320M described in (13) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00249]~~[00250] For example, an RU52 is used as an example. An 80 MHz sequence includes 16 RU52s, that is, 16 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 16 * 4 = 64 PAPRs may be obtained by considering rotated sequences. A maximum value of the 64 PAPRs is 4.95. 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M are obtained based on the sequence 4xEHT_LTF_80M. PAPR values of 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M on the RU52s are the same as the PAPR values of 4xEHT_LTF_80M on the RU52s.

~~[00250]~~[00251] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00251]~~[00252] It may be learned from Table 10 that values in the second column are all less

than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 10: Comparison between PAPR maximum values of 4x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR	Median PAPR of BPSK Data
RU26	4.49	5.89
RU52	4.95	6.71
RU52 + RU26	5.79	7.10
RU106	5.13	7.29
RU106 + RU26	5.33	7.44
RU242	6.12	7.94
RU484	6.41	8.44
RU484 + RU242	7.95	8.83
RU996	6.34	8.84
RU996 + RU484	8.79	9.17
RU2*996	6.32	9.27 or 9.28
RU2*996 + RU484	8.91	9.50
RU3*996	8.29	9.54
RU3*996 + RU484	9.29	9.55
RU4*996	8.97	9.60

~~[00252]~~[00253] (16) A possible 4x LTF sequence with 80 MHz bandwidth is denoted by

4xEHT_LTF_80M. Subcarrier numbers of the sequence 4xEHT_LTF_80M range from –500 to 500.

~~[00253]~~[00254] For example, $4xEHT_LTF_80M_{-500:500}$ = {4xEHT_LTF_partA, 0_5 , 4xEHT_LTF_partB}.

~~[00254]~~[00255] Herein, –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 4xEHT_LTF_partA includes 498 elements, and 4xEHT_LTF_partB includes 498 elements.

~~[00255]~~[00256] In an example, 4xEHT_LTF_partA =

{–1 1 1 1 1 1 –1 –1 –1 1 –1 –1 1 –1
–1 1 1 –1 –1 –1 –1 1 –1 1 –1 1 –1 1
–1 1 –1 1 1 –1 1 1 1 –1 –1 –1 1 1
–1 –1 1 –1 1 1 1 1 1 1 1 –1 –1 –1
–1 –1 –1 –1 1 1 1 –1 1 1 –1 1 1 –1
–1 1 1 1 1 –1 1 –1 1 –1 1 1 –1 1
–1 1 1 –1 1 1 1 –1 –1 –1 1 1 –1 –1
1 –1 1 1 1 1 1 1 1 1 1 –1 1 1
–1 1 –1 1 1 1 1 –1 –1 1 1 –1 –1 1
–1 1 1 1 1 1 –1 –1 –1 1 1 1 1 1
–1 –1 –1 1 –1 –1 1 –1 –1 1 1 –1 –1 –1
–1 1 –1 1 –1 1 –1 1 –1 1 –1 1 1 –1
1 1 1 –1 –1 –1 1 1 –1 –1 1 –1 1 1
1 1 1 1 1 –1 –1 1 1 1 1 1 –1 –1
–1 1 –1 –1 1 –1 –1 1 1 –1 –1 –1 –1 1
–1 1 –1 1 –1 –1 1 –1 1 –1 –1 1 –1 –1
–1 1 1 1 –1 –1 1 1 –1 1 –1 –1 –1 –1
–1 –1 –1 1 1 –1 –1 1 1 1 1 –1 1 –1
1 –1 1 1 1 1 –1 –1 1 1 –1 1 1 –1
1 1 1 –1 –1 –1 –1 –1 1 1 1 1 1 1
1 –1 1 –1 –1 1 1 –1 –1 –1 1 1 1 –1
1 1 –1 1 –1 1 1 –1 –1 1 –1 1 –1 1
–1 –1 –1 –1 1 1 –1 –1 1 –1 –1 1 –1 –1

–1 1 1 1 1 1 1 1 1 1 1 1 –1

1 –1 –1 1 1 –1 –1 –1 1 1 1 –1 1 1

–1 1 –1 1 1 –1 1 1 –1 1 –1 1 1 1

1 –1 –1 1 1 –1 –1 1 –1 1 1 1 1 1

–1 –1 –1 1 –1 1 –1 1 –1 1 –1 –1 –1 –1

1 1 –1 –1 1 –1 –1 1 –1 –1 –1 1 1 1

1 1 –1 –1 –1 –1 –1 –1 –1 1 –1 1 1 –1

–1 1 1 1 –1 –1 –1 1 –1 –1 1 –1 1 –1

1 –1 –1 1 –1 1 –1 1 –1 –1 –1 –1 1 1

–1 –1 1 –1 –1 1 –1 –1 –1 1 1 1 1 1

1 1 1 1 1 1 1 –1 1 –1 –1 1 1 –1

–1 –1 1 1 1 –1 1 1 –1 1 –1 1 1 –1

1 1 –1 –1 1 1 1 1}

4xEHT_LTF_partB =

{–1 –1 –1 1 –1 1 –1 1 1 –1 1 1 1 1

1 –1 –1 –1 1 –1 –1 1 –1 –1 1 1 –1 –1

–1 –1 1 –1 1 –1 1 –1 1 –1 1 –1 1 1

–1 1 1 1 –1 –1 –1 1 1 –1 –1 1 –1 1

1 1 1 1 1 1 –1 1 –1 –1 –1 –1 –1 1

1 1 –1 1 1 –1 1 1 –1 –1 1 1 1 1

–1 1 –1 1 –1 1 1 –1 1 –1 1 1 –1 1

1 1 –1 –1 –1 1 1 –1 –1 1 –1 1 1 1

1 1 1 1 1 1 –1 1 1 –1 1 –1 1 1

1 –1 –1 –1 1 1 1 1 1 –1 –1 –1 1 –1

–1 1 –1 –1 1 1 –1 –1 –1 –1 1 –1 1 –1

1 –1 1 –1 1 –1 1 1 –1 1 1 1 –1 –1

–1 1 1 –1 –1 1 –1 1 1 1 1 1 1 1

–1 –1 1 1 1 1 1 –1 –1 –1 1 –1 –1 1

–1 –1 1 1 –1 –1 –1 –1 1 –1 1 –1 1 –1

–1 1 –1 1 –1 –1 1 –1 –1 –1 1 1 1 –1

–1 1 1 –1 1 –1 –1 –1 –1 –1 –1 –1 1 –1

1 –1 –1 1 –1 –1 1 –1 1 –1 1 –1 –1 –1

–1 1 1 –1 –1 1 –1 –1 1 –1 –1 –1 1 1

1 1 1 –1 –1 –1 –1 –1 –1 –1 1 –1 1 1

–1 –1 1 1 1 –1 –1 –1 1 –1 –1 1 –1 1

–1 –1 1 1 –1 1 –1 1 –1 1 1 1 1 –1

–1 1 1 –1 1 1 –1 1 1 1 –1 –1 –1 –1

–1 –1 –1 –1 –1 –1 –1 –1 1 –1 1 1 –1 –1

1 1 1 –1 –1 –1 1 –1 –1 1 –1 1 –1 –1

1 –1 –1 1 –1 1 –1 –1 –1 –1 1 1 –1 –1

1 1 –1 1 –1 –1 –1 –1 –1 1 1 1 –1 1

–1 1 –1 1 –1 1 1 1 1 –1 –1 1 1 –1

1 1 –1 1 1 1 –1 –1 –1 –1 –1 1 1 1

1 1 1 1 –1 1 –1 –1 1 1 –1 –1 –1 1

1 1 –1 1 1 –1 1 –1 1 –1 1 1 –1 1

–1 1 –1 1 1 1 1 –1 –1 1 1 –1 1 1

–1 1 1 1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1

–1 –1 1 –1 1 1 –1 –1 1 1 1 –1 –1 –1

1 –1 –1 1 –1 1 –1 –1}

~~[00256]~~[00257] (17) A possible 4x LTF sequence with 160 MHz bandwidth is denoted by 4xEHT_LTF_160M. 4xEHT_LTF_160M may be constructed based on 4xEHT_LTF_80M described in (16), and subcarrier numbers of the sequence 4xEHT_LTF_160M range from –1012 to 1012.

~~[00257]~~[00258] For example, $4xEHT_LTF_160M_{-1012:1012}$ =

{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, –4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}.

~~[00258]~~[00259] Herein, –4xEHT_LTF_partA represents negation (multiplied by –1) of all elements in the sequence 4xEHT_LTF_partA; 0_{23} represents 23 consecutive 0s; 0_5 represents five consecutive 0s; and 4xEHT_LTF_partA and 4xEHT_LTF_partB in $4xEHT_LTF_160M_{-1012:1012}$ are 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (16).

~~[00259]~~[00260] (18) A possible 4x LTF sequence with 320 MHz bandwidth is denoted by

4xEHT_LTF_320M. 4xEHT_LTF_320M is constructed based on 4xEHT_LTF_160M described in (17), and subcarrier numbers of the sequence 4xEHT_LTF_320M range from –2036 to 2036.

~~[00260]~~**[00261]** For example, $4xEHT_LTF_320M_{-2036:2036}$ =

$\{4xEHT_LTF_160M_{-1012:1012}, 0_{23}, -4xEHT_LTF_80M_{-500:500}, 0_{23}, -4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB\}$.

~~[00261]~~**[00262]** Herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; –4xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 4xEHT_LTF_partA; 0_{23} represents 23 consecutive 0s; and 0_5 represents five consecutive 0s. In $4xEHT_LTF_320M_{-2036:2036}$, $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (17), 4xEHT_LTF_partA and 4xEHT_LTF_partB are 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (16), and $4xEHT_LTF_80M_{-500:500}$ is $4xEHT_LTF_80M_{-500:500}$ described in (16).

~~[00262]~~**[00263]** (19) A possible 4x LTF sequence with 240 MHz bandwidth is denoted by 4xEHT_LTF_240M. 4xEHT_LTF_240M is constructed based on 4xEHT_LTF_160M described in (17) and 4xEHT_LTF_80M described in (16). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 4x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 4xEHT_LTF_240M range from –1524 to 1524.

~~[00263]~~**[00264]** For example, $4xEHT_LTF_240M_{-1524:1524}$ =

$\{4xEHT_LTF_160M_{-1012:1012}, 0_{23}, -4xEHT_LTF_80M_{-500:500}\}$.

~~[00264]~~**[00265]** Herein, $-4xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[00265]~~**[00266]** In $4xEHT_LTF_240M_{-1524:1524}$ in the two examples in (19), $4xEHT_LTF_80M_{-500:500}$ is $4xEHT_LTF_80M_{-500:500}$ in (16), and $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (17).

~~[00266]~~**[00267]** (20) Sequences obtained by performing one or more of the following operations on the 4x LTF sequences with various bandwidth described in (16) to (19), for example, $4xEHT_LTF_80M_{-500:500}$, $4xEHT_LTF_160M_{-1012:1012}$, $4xEHT_LTF_240M_{-1524:1524}$, and $4xEHT_LTF_320M_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on

entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00267]~~[00268] Operation (1): Multiply elements in a sequence by –1.

~~[00268]~~[00269] Operation (2): Reverse an order of elements in a sequence.

~~[00269]~~[00270] Operation (3): Multiply an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[00270]~~[00271] As shown in Table 11 below, Table 11 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 4xEHT_LTF_320M described in (18) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00271]~~[00272] For example, an RU52 is used as an example. An 80 MHz sequence includes 16 RU52s, that is, 16 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 16 * 4 = 64 PAPRs may be obtained by considering rotated sequences. A maximum value of the 64 PAPRs is 5.73. 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M are obtained based on the sequence 4xEHT_LTF_80M. PAPR values of 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M on the RU52s are the same as the PAPR values of 4xEHT_LTF_80M on the RU52s.

~~[00272]~~[00273] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00273]~~[00274] It may be learned from Table 11 that most of values in the second column are less than values in the third column. That is, most of maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are less than corresponding PAPR median values of a BPSK data part. In addition, PAPRs are particularly low for important RUs such as RU4*996, RU2*996, and RU996. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences

have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 11: Comparison between PAPR maximum values of 4x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR	Median PAPR of BPSK Data
RU26	4.04	5.89
RU52	5.73	6.71
RU52 + RU26	6.65	7.10
RU106	6.05	7.29
RU106 + RU26	6.33	7.44
RU242	5.70	7.94
RU484	6.09	8.44
RU484 + RU242	8.25	8.83
RU996	6.51	8.84
RU996 + RU484	8.30	9.17
RU2*996	6.38	9.27 or 9.28
RU2*996 + RU484	9.57	9.50
RU3*996	8.69	9.54
RU3*996 + RU484	8.85	9.55
RU4*996	6.41	9.60

~~[00274]~~**[00275]** This embodiment of this application further provides the following LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode.

~~[00275]~~**[00276]** (21) A possible 2x LTF sequence with 80 MHz bandwidth is denoted by 2xEHT_LTF_80M. Subcarrier numbers of the sequence 2xEHT_LTF_80M range from –500 to 500.

~~[00276]~~**[00277]** For example, $2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB}, where –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 2xEHT_LTF_partA includes 498 elements, and

2xEHT_LTF_partB includes 498 elements.

~~[00277]~~[00278] In an example, 2xEHT_LTF_partA =

{–1 0 –1 0 –1 0 –1 0 –1 0 1 0 1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
–1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
1 0 1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 –1 0 1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 –1 0
–1 0 1 0 1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 1 0 –1 0 1 0 1 0 1 0
1 0 1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 1 0
–1 0 –1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 –1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 –1 0 1 0 1 0 –1 0 –1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 –1 0 1 0 1 0 1 0
–1 0 –1 0 1 0 1 0 –1 0 1 0 1 0
–1 0 1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 1 0 1 0 1 0
1 0 1 0 1 0 –1 0 1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 1 0 1 0
–1 0 –1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 –1 0
1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

1 0 1 0 –1 0 1 0 1 0 –1 0 –1 0

1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 1 0 –1 0 –1 0 –1 0 1 0

1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

–1 0 –1 0 1 0 1 0 1 0 1 0 1 0

1 0 –1 0 –1 0 1 0}

2xEHT_LTF_partB =

{0 –1 0 1 0 –1 0 –1 0 –1 0 –1 0 –1

0 –1 0 –1 0 1 0 1 0 1 0 –1 0 1

0 –1 0 1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 1 0 –1 0 1 0 1 0 –1

0 –1 0 1 0 –1 0 –1 0 1 0 –1 0 1

0 –1 0 1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 –1 0 1 0 1 0 1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 –1 0 1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 –1 0 1 0 –1 0 –1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 1 0 –1 0 –1 0 1 0 1 0 1

0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 –1 0 1

0 1 0 1 0 1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 –1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 –1 0 1 0 1 0 1

0 1 0 –1 0 –1 0 1 0 1 0 1 0 –1

0 1 0 1 0 1 0 1 0 –1 0 –1 0 1

0 –1 0 1 0 –1 0 1 0 –1 0 –1 0 1

0 –1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 1 0 1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 1

0 1 0 1 0 1 0 1 0 –1 0 –1 0 1

0 –1 0 1 0 –1 0 1 0 1 0 1 0 –1

0 –1 0 –1 0 –1 0 –1}

~~[00278]~~[00279] (22) A possible 2x LTF sequence with 160 MHz bandwidth is denoted by 2xEHT_LTF_160M. 2xEHT_LTF_160M may be constructed based on 2xEHT_LTF_partA and 2xEHT_LTF_partB described in (21), and subcarrier numbers of the sequence 2xEHT_LTF_160M range from –1012 to 1012.

~~[00279]~~[00280] For example, $2xEHT_LTF_160M_{-1012:1012}$ =

{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, –2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}.

~~[00280]~~[00281] Herein, 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (21); –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00281]~~[00282] (23) A possible 2x LTF sequence with 320 MHz bandwidth is denoted by 2xEHT_LTF_320M. 2xEHT_LTF_320M is constructed based on 2xEHT_LTF_partA and 2xEHT_LTF_partB described in (21), and subcarrier numbers of the sequence 2xEHT_LTF_320M range from –2036 to 2036.

~~[00282]~~[00283] For example, $2xEHT_LTF_320M_{-2036:2036}$ =

{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, –

2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}.

~~[00283]~~[00284] Herein, 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (21); –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00284]~~[00285] (24) A possible 2x LTF sequence with 240 MHz bandwidth is denoted by 2xEHT_LTF_240M. 2xEHT_LTF_240M is constructed based on 2xEHT_LTF_160M described in (22) or based on 2xEHT_LTF_80M described in (21). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 2x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 2xEHT_LTF_240M range from –1524 to 1524.

~~[00285]~~[00286] For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{2xEHT_LTF_80M, 0_{23}, 2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_80M}.

~~[00286]~~[00287] Herein, 2xEHT_LTF_80M is 2xEHT_LTF_80M$_{-500:500}$ in (21); 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (21); –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00287]~~[00288] For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_160M$_{-1012:1012}$}.

~~[00288]~~[00289] Herein, 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ in (22); 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (21); –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00289]~~[00290] (25) Sequences obtained by performing one or more of the following operations on the 2x LTF sequences with various bandwidth described in (21) to (24), for example, 2xEHT_LTF_80M$_{-500:500}$, 2xEHT_LTF_160M$_{-1012:1012}$, 2xEHT_LTF_240M$_{-1524:1524}$, and

2xEHT_LTF_320M$_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00290]~~[00291] Operation (1): Multiply elements in a sequence by –1.

~~[00291]~~[00292] Operation (2): Reverse an order of elements in a sequence. It is assumed that a sequence is 123456, and a sequence obtained by reversing an order of elements in the sequence is 654321.

~~[00292]~~[00293] Operation (3): Multiply an even-numbered or odd-numbered element in non-zero elements by –1.

~~[00293]~~[00294] As shown in Table 12 below, Table 12 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 2xEHT_LTF_320M described in (23) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00294]~~[00295] For example, an RU26 is used as an example. An 80 MHz sequence includes 36 RU26s, that is, 36 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 36 * 4 = 144 PAPRs may be obtained by considering rotated sequences. A maximum value of the 144 PAPRs is 5.76. 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M are obtained based on the sequence 2xEHT_LTF_80M. PAPR values of 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M on the RU26s are the same as the PAPR values of 2xEHT_LTF_80M on the RU26s.

~~[00295]~~[00296] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00296]~~[00297] It may be learned from Table 12 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode that are generated in this application have relatively low PAPR values

on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 12: Comparison between PAPR maximum values of 2x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU26	5.76	5.89
RU52	4.98	6.71
RU52 + RU26	6.23	7.10
RU106	4.55	7.29
RU106 + RU26	5.45	7.44
RU242	5.09	7.94
RU484	5.53	8.44
RU484 + RU242	7.61	8.83
RU996	5.63	8.84
RU996 + RU484	7.74	9.17
RU2*996	5.62 or 8.53	9.27 or 9.28
RU2*996 + RU484	8.99	9.50
RU3*996	7.93	9.54
RU3*996 + RU484	7.91	9.55
RU4*996	5.75	9.60

~~[00297]~~[00298] (26) A possible 2x LTF sequence with 80 MHz bandwidth is denoted by 2xEHT_LTF_80M. Subcarrier numbers of the sequence 2xEHT_LTF_80M range from –500 to 500.

~~[00298]~~[00299] For example, $2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB}, where –500 to 500 are subcarrier indexes (the indexes may also be referred

to as numbers), 0_5 represents five consecutive 0s, 2xEHT_LTF_partA includes 498 elements, and 2xEHT_LTF_partB includes 498 elements.

~~[00299]~~[00300] In an example, 2xEHT_LTF_partA =

{–1 0 –1 0 –1 0 –1 0 –1 0 1 0 1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
–1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
1 0 1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 –1 0 1 0 1 0
1 0 –1 0 –1 0 1 0 1 0 1 0 1 0
–1 0 1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 –1 0 1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 –1 0 1 0 1 0
1 0 1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 –1 0 1 0 1 0
–1 0 –1 0 –1 0 1 0 1 0 1 0 1 0
1 0 1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 1 0 –1 0 1 0 1 0 –1 0 –1 0
1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 –1 0 1 0 1 0 1 0
–1 0 –1 0 1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 –1 0 1 0 1 0
1 0 1 0 1 0 –1 0 1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 1 0
1 0 1 0 –1 0 1 0 1 0 1 0 1 0

–1 0 –1 0 1 0 –1 0 1 0 –1 0 1 0

–1 0 –1 0 1 0 –1 0 –1 0 1 0 1 0

–1 0 1 0 1 0 1 0 1 0 1 0 1 0

–1 0 –1 0 –1 0 1 0 1 0 1 0 –1 0

–1 0 1 0 1 0 1 0 1 0 1 0 1 0

–1 0 –1 0 1 0 –1 0 1 0 –1 0 1 0

1 0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0

–1 0 1 0 1 0 –1 0}

2xEHT_LTF_partB =

{0 –1 0 1 0 1 0 –1 0 –1 0 –1 0 –1

0 –1 0 –1 0 1 0 1 0 1 0 –1 0 1

0 –1 0 1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 1 0 –1 0 1 0 1 0 –1

0 –1 0 1 0 –1 0 –1 0 1 0 –1 0 1

0 –1 0 1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 –1 0 –1 0 1 0 1 0 –1 0 –1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 –1 0 1 0 1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 –1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 1 0 1 0 1 0 1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 1 0 1

0 –1 0 1 0 1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 –1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 –1 0 –1 0 1 0 1 0 1 0 –1

0 –1 0 1 0 1 0 –1 0 1 0 –1 0 1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 1 0 1 0 –1

0 1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 –1 0 –1 0 –1 0 1 0 1 0 –1 0 –1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 –1 0 1

0 1 0 1 0 1 0 1}

~~[00300]~~[00301] (27) A possible 2x LTF sequence with 160 MHz bandwidth is denoted by 2xEHT_LTF_160M. 2xEHT_LTF_160M may be constructed based on 2xEHT_LTF_partA and 2xEHT_LTF_partB described in (26), and subcarrier numbers of the sequence 2xEHT_LTF_160M range from –1012 to 1012.

~~[00301]~~[00302] For example, 2xEHT_LTF_160M$_{-1012:1012}$ =

{–2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}.

~~[00302]~~[00303] Herein, 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (26); –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00303]~~[00304] (28) A possible 2x LTF sequence with 320 MHz bandwidth is denoted by 2xEHT_LTF_320M. 2xEHT_LTF_320M is constructed based on 2xEHT_LTF_160M described in (27), and subcarrier numbers of the sequence 2xEHT_LTF_320M range from –2036 to 2036.

~~[00304]~~[00305] For example, 2xEHT_LTF_320M$_{-2036:2036}$ =

{–2xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, 2xEHT_LTF_160M$_{-1012:1012}$}.

~~[00305]~~[00306] Herein, 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ in (27); –

2xEHT_LTF_160M$_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_160M$_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[00306]~~**[00307]** (29) A possible 2x LTF sequence with 240 MHz bandwidth is denoted by 2xEHT_LTF_240M. 2xEHT_LTF_240M may be constructed based on 2xEHT_LTF_160M described in (27) or based on 2xEHT_LTF_80M described in (26). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 2x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 2xEHT_LTF_240M range from –1524 to 1524.

~~[00307]~~**[00308]** For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{–2xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, –2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}.

~~[00308]~~**[00309]** Herein, 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ described in (27); 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (26); –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; –2xEHT_LTF_160M$_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_160M$_{-1012:1012}$; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00309]~~**[00310]** For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{–2xEHT_LTF_80M$_{-500:500}$, 0_{23}, 2xEHT_LTF_160M$_{-1012:1012}$}.

~~[00310]~~**[00311]** Herein, 2xEHT_LTF_80M$_{-500:500}$ is 2xEHT_LTF_80M$_{-500:500}$ in (26); 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ in (27); –2xEHT_LTF_80M$_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_80M$_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[00311]~~**[00312]** (30) Sequences obtained by performing one or more of the following operations on the 2x LTF sequences with various bandwidth described in (26) to (29), for example, 2xEHT_LTF_80M$_{-500:500}$, 2xEHT_LTF_160M$_{-1012:1012}$, 2xEHT_LTF_240M$_{-1524:1524}$, and 2xEHT_LTF_320M$_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00312]~~**[00313]** Operation (1): Multiply elements in a sequence by –1.

~~[00313]~~**[00314]** Operation (2): Reverse an order of elements in a sequence. It is assumed that a

sequence is 123456, and a sequence obtained by reversing an order of elements in the sequence is 654321.

~~[00314]~~[00315] Operation (3): Multiply an even-numbered or odd-numbered element in non-zero elements by –1.

~~[00315]~~[00316] As shown in Table 13 below, Table 13 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 2xEHT_LTF_320M described in (28) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00316]~~[00317] For example, an RU26 is used as an example. An 80 MHz sequence includes 36 RU26s, that is, 36 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 36 * 4 = 144 PAPRs may be obtained by considering rotated sequences. A maximum value of the 144 PAPRs is 5.76. 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M are obtained based on the sequence 2xEHT_LTF_80M. PAPR values of 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M on the RU26s are the same as the PAPR values of 2xEHT_LTF_80M on the RU26s.

~~[00317]~~[00318] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00318]~~[00319] It may be learned from Table 13 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 13: Comparison between PAPR maximum values of 2x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU26	5.76	5.89
RU52	4.98	6.71
RU52 + RU26	6.23	7.10
RU106	4.55	7.29
RU106 + RU26	5.45	7.44
RU242	5.09	7.94
RU484	5.53	8.44
RU484 + RU242	7.61	8.83
RU996	5.69	8.84
RU996 + RU484	7.54	9.17
RU2*996	6.01 or 8.58	9.27 or 9.28
RU2*996 + RU484	8.31	9.50
RU3*996	7.69	9.54
RU3*996 + RU484	8.28	9.55
RU4*996	8.33	9.60

~~[00319]~~[00320] (31) A possible 2x LTF sequence with 80 MHz bandwidth is denoted by 2xEHT_LTF_80M. Subcarrier numbers of the sequence 2xEHT_LTF_80M range from –500 to 500.

~~[00320]~~[00321] For example, $2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB}, where –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 2xEHT_LTF_partA includes 498 elements, and 2xEHT_LTF_partB includes 498 elements.

~~[00321]~~[00322] For example, the sequence 2xEHT_LTF_partB is obtained by reversing an order of the sequence 2xEHT_LTF_partA and then negating an even-numbered element in non-zero elements, that is, negating an element at a position that is an integer multiple of 4.

~~[00322]~~[00323] In an example, 2xEHT_LTF_partA =

{1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0

1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 1 0 –1 0 1 0 –1 0 1 0

1 0 1 0 –1 0 1 0 1 0 1 0 1 0

1 0 1 0 1 0 –1 0 –1 0 –1 0 1 0

1 0 –1 0 1 0 –1 0 1 0 –1 0 1 0

1 0 –1 0 1 0 1 0 1 0 1 0 –1 0

–1 0 1 0 1 0 –1 0 1 0 –1 0 –1 0

–1 0 –1 0 –1 0 –1 0 1 0 –1 0 1 0

1 0 –1 0 –1 0 1 0 –1 0 –1 0 1 0

1 0 –1 0 –1 0 –1 0 1 0 1 0 –1 0

1 0 –1 0 1 0 –1 0 –1 0 1 0 –1 0

–1 0 –1 0 –1 0 –1 0 1 0 –1 0 –1 0

–1 0 1 0 –1 0 1 0 1 0 1 0 1 0

–1 0 1 0 –1 0 –1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0

1 0 –1 0 1 0 1 0 –1 0 –1 0 1 0

1 0 1 0 1 0 –1 0 1 0 1 0 1 0

1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0

–1 0 –1 0 –1 0 1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0

–1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

–1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1 0

–1 0 1 0 –1 0 –1 0 1 0 –1 0 1 0

–1 0 1 0 1 0 –1 0 1 0 –1 0 1 0

1 0 –1 0 1 0 1 0 1 0 1 0 –1 0

–1 0 –1 0 1 0 –1 0 –1 0 1 0 1 0

1 0 –1 0 –1 0 –1 0 –1 0 1 0 –1 0

–1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

–1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0

–1 0 –1 0 –1 0 –1 0 1 0 –1 0 –1 0

–1 0 1 0 –1 0 1 0 –1 0 –1 0 –1 0

1 0 1 0 1 0 1 0 1 0 1 0 –1 0

–1 0 1 0 –1 0 –1 0 1 0 1 0 –1 0

–1 0 1 0 –1 0 –1 0}

2xEHT_LTF_partB =

{0 –1 0 –1 0 1 0 1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 –1 0 1 0 1

0 1 0 1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 –1 0 1

0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 1 0 –1

0 –1 0 –1 0 –1 0 1 0 1 0 –1 0 –1

0 1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 1 0 1 0 –1 0 –1 0 –1 0 1 0 1

0 1 0 1 0 –1 0 1 0 1 0 –1 0 –1

0 –1 0 1 0 1 0 –1 0 1 0 –1 0 1

0 –1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1

0 –1 0 1 0 –1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 –1 0 1 0 1 0 1 0 1

0 –1 0 1 0 –1 0 –1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 1 0 1

0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 1 0 –1 0 1

0 –1 0 1 0 –1 0 1 0 1 0 1 0 –1

0 1 0 1 0 1 0 1 0 1 0 –1 0 1

0 1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 1

0 –1 0 1 0 1 0 –1 0 –1 0 1 0 –1

0 1 0 1 0 1 0 1 0 1 0 1 0 –1

0 1 0 –1 0 –1 0 1 0 1 0 –1 0 –1

0 –1 0 –1 0 1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 1 0 1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 –1 0 1 0 –1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 1 0 1 0 1 0 –1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 –1}

~~[00323]~~[00324] (32) A possible 2x LTF sequence with 160 MHz bandwidth is denoted by 2xEHT_LTF_160M. 2xEHT_LTF_160M may be constructed based on 2xEHT_LTF_partA, 2xEHT_LTF_partB, and 2xEHT_LTF_80M$_{-500:500}$ described in (31), and subcarrier numbers of the sequence 2xEHT_LTF_160M range from –1012 to 1012.

~~[00324]~~[00325] For example, 2xEHT_LTF_160M$_{-1012:1012}$ =

{2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_80M$_{-500:500}$}.

~~[00325]~~[00326] Herein, 2xEHT_LTF_80M$_{-500:500}$, 2xEHT_LTF_partA, and 2xEHT_LTF_partB are respectively 2xEHT_LTF_80M$_{-500:500}$, 2xEHT_LTF_partA, and 2xEHT_LTF_partB in (31); –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00326]~~[00327] (33) A possible 2x LTF sequence with 320 MHz bandwidth is denoted by 2xEHT_LTF_320M. 2xEHT_LTF_320M is constructed based on 2xEHT_LTF_partA and 2xEHT_LTF_partB described in (31), and subcarrier numbers of the sequence 2xEHT_LTF_320M range from –2036 to 2036.

~~[00327]~~[00328] For example, 2xEHT_LTF_320M$_{-2036:2036}$ =

{2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB, 0_{23}, –2xEHT_LTF_partA, 0_5, –2xEHT_LTF_partB}.

~~[00328]~~[00329] Herein, 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (31); –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of

all elements in the sequence 2xEHT_LTF_partA; –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00329]~~[00330] (34) A possible 2x LTF sequence with 240 MHz bandwidth is denoted by 2xEHT_LTF_240M. 2xEHT_LTF_240M is constructed based on 2xEHT_LTF_160M described in (32) or based on 2xEHT_LTF_partA and 2xEHT_LTF_partB in (31). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 2x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 2xEHT_LTF_240M range from –1524 to 1524.

~~[00330]~~[00331] For example, $2xEHT_LTF_240M_{-1524:1524}$ =

$\{2xEHT_LTF_160M_{-1012:1012}, 0_{23}, 2xEHT_LTF_partA, 0_5, -2xEHT_LTF_partB\}$.

~~[00331]~~[00332] Herein, $2xEHT_LTF_160M_{-1012:1012}$ is $2xEHT_LTF_160M_{-1012:1012}$ in (32); 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (31); –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00332]~~[00333] For example, $2xEHT_LTF_240M_{-1524:1524}$ =

$\{2xEHT_LTF_80M_{-500:500}, 0_{23}, 2xEHT_LTF_partA, 0_5, -2xEHT_LTF_partB, 0_{23}, -2xEHT_LTF_80M_{-500:500}\}$.

~~[00333]~~[00334] Herein, $2xEHT_LTF_80M_{-500:500}$, 2xEHT_LTF_partA, and 2xEHT_LTF_partB are respectively $2xEHT_LTF_80M_{-500:500}$, 2xEHT_LTF_partA, and 2xEHT_LTF_partB described in (31); $-2xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $2xEHT_LTF_80M_{-500:500}$; –2xEHT_LTF_partB represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partB; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00334]~~[00335] (35) Sequences obtained by performing one or more of the following operations on the 2x LTF sequences with various bandwidth described in (31) to (34), for example, $2xEHT_LTF_80M_{-500:500}$, $2xEHT_LTF_160M_{-1012:1012}$, $2xEHT_LTF_240M_{-1524:1524}$, and $2xEHT_LTF_320M_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on

entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00335]~~[00336] Operation (1): Multiply elements in a sequence by –1.

~~[00336]~~[00337] Operation (2): Reverse an order of elements in a sequence. It is assumed that a sequence is 123456, and a sequence obtained by reversing an order of elements in the sequence is 654321.

~~[00337]~~[00338] Operation (3): Multiply an even-numbered or odd-numbered element in non-zero elements by –1.

~~[00338]~~[00339] As shown in Table 14 below, Table 14 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 2xEHT_LTF_320M described in (33) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00339]~~[00340] For example, an RU26 is used as an example. An 80 MHz sequence includes 36 RU26s, that is, 36 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 36 * 4 = 144 PAPRs may be obtained by considering rotated sequences. A maximum value of the 144 PAPRs is 5.96. 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M are obtained based on the sequence 2xEHT_LTF_80M. PAPR values of 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M on the RU26s are the same as the PAPR values of 2xEHT_LTF_80M on the RU26s.

~~[00340]~~[00341] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00341]~~[00342] It may be learned from Table 14 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences

obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 14: Comparison between PAPR maximum values of 2x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU26	5.96	5.89
RU52	4.99	6.71
RU52 + RU26	6.23	7.10
RU106	4.55	7.29
RU106 + RU26	5.45	7.44
RU242	5.09	7.94
RU484	5.53	8.44
RU484 + RU242	7.61	8.83
RU996	5.45	8.84
RU996 + RU484	7.71	9.17
RU2*996	5.63 or 8.37	9.27 or 9.28
RU2*996 + RU484	8.37	9.50
RU3*996	7.85	9.54
RU3*996 + RU484	7.90	9.55
RU4*996	5.72	9.60

~~[00342]~~[00343] (36) A possible 2x LTF sequence with 80 MHz bandwidth is denoted by 2xEHT_LTF_80M. Subcarrier numbers of the sequence 2xEHT_LTF_80M range from –500 to 500.

~~[00343]~~[00344] For example, $2xEHT_LTF_80M_{-500:500}$ = {2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB}, where –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 2xEHT_LTF_partA includes 498 elements, and 2xEHT_LTF_partB includes 498 elements.

~~[00344]~~[00345] For example, the sequence 2xEHT_LTF_partB is obtained by reversing an order of the sequence 2xEHT_LTF_partA and then negating an even-numbered element in non-zero elements, that is, negating an element at a position that is an integer multiple of 4.

~~[00345]~~[00346] In an example, 2xEHT_LTF_partA =

{–1 0 –1 0 –1 0 –1 0 –1 0 1 0 1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
–1 0 1 0 1 0 1 0 –1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 1 0 1 0 –1 0
1 0 1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 1 0 –1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 1 0 –1 0 1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 –1 0
–1 0 1 0 1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 1 0 –1 0 1 0 1 0 1 0
1 0 1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 1 0
–1 0 –1 0 1 0 –1 0 –1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 1 0 –1 0 –1 0
1 0 1 0 1 0 –1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 –1 0 –1 0 1 0 1 0 –1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 –1 0 –1 0
–1 0 –1 0 –1 0 –1 0 1 0 1 0 1 0
–1 0 –1 0 1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 1 0 –1 0 –1 0
1 0 –1 0 –1 0 1 0 1 0 –1 0 1 0
–1 0 1 0 –1 0 –1 0 1 0 1 0 1 0
1 0 1 0 1 0 –1 0 1 0 1 0 –1 0
1 0 –1 0 1 0 1 0 –1 0 1 0 1 0
–1 0 –1 0 1 0 1 0 1 0 1 0 –1 0

1 0 –1 0 1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

1 0 1 0 –1 0 1 0 1 0 –1 0 –1 0

1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 1 0 –1 0 –1 0 –1 0 1 0

1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0

1 0 1 0 –1 0 1 0 –1 0 1 0 –1 0

–1 0 –1 0 1 0 1 0 1 0 1 0 1 0

1 0 –1 0 –1 0 1 0}

2xEHT_LTF_partB =

{0 1 0 –1 0 1 0 1 0 1 0 1 0 1

0 1 0 1 0 –1 0 –1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 1 0 –1 0 –1 0 –1

0 –1 0 –1 0 –1 0 1 0 1 0 –1 0 –1

0 –1 0 1 0 1 0 1 0 –1 0 –1 0 –1

0 –1 0 –1 0 –1 0 1 0 –1 0 –1 0 1

0 1 0 –1 0 1 0 1 0 –1 0 1 0 –1

0 1 0 –1 0 1 0 1 0 –1 0 –1 0 –1

0 –1 0 1 0 –1 0 –1 0 –1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 1 0 –1

0 –1 0 –1 0 1 0 –1 0 1 0 –1 0 –1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 –1 0 1 0 1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 –1 0 1 0 1 0 –1 0 –1 0 –1

0 1 0 1 0 1 0 1 0 1 0 1 0 1

0 –1 0 –1 0 1 0 –1 0 1 0 1 0 –1

0 –1 0 –1 0 –1 0 –1 0 –1 0 –1 0 1

0 1 0 –1 0 –1 0 –1 0 –1 0 –1 0 –1

0 –1 0 1 0 1 0 1 0 –1 0 –1 0 1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 1 0 1 0 –1 0 1 0 –1 0 1 0 –1

0 –1 0 1 0 1 0 1 0 1 0 1 0 1

0 –1 0 1 0 1 0 –1 0 1 0 –1 0 1

0 1 0 –1 0 –1 0 1 0 –1 0 –1 0 –1

0 –1 0 1 0 1 0 –1 0 –1 0 –1 0 1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 1 0 1 0 –1

0 1 0 1 0 –1 0 –1 0 1 0 –1 0 –1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 1

0 –1 0 –1 0 –1 0 1 0 1 0 –1 0 –1

0 –1 0 –1 0 –1 0 –1 0 1 0 1 0 –1

0 1 0 –1 0 1 0 –1 0 –1 0 –1 0 1

0 1 0 1 0 1 0 1}

~~[00346]~~[00347] (37) A possible 2x LTF sequence with 160 MHz bandwidth is denoted by 2xEHT_LTF_160M. 2xEHT_LTF_160M may be constructed based on 2xEHT_LTF_partA, 2xEHT_LTF_partB, and 2xEHT_LTF_80M$_{–500:500}$ described in (36), and subcarrier numbers of the sequence 2xEHT_LTF_160M range from –1012 to 1012.

~~[00347]~~[00348] For example, 2xEHT_LTF_160M$_{–1012:1012}$ =

{–2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_80M$_{–500:500}$}.

~~[00348]~~[00349] Herein, 2xEHT_LTF_partA, 2xEHT_LTF_partB, and 2xEHT_LTF_80M$_{–500:500}$ are respectively 2xEHT_LTF_partA, 2xEHT_LTF_partB, and 2xEHT_LTF_80M$_{–500:500}$ in (36); –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00349]~~[00350] (38) A possible 2x LTF sequence with 320 MHz bandwidth is denoted by 2xEHT_LTF_320M. 2xEHT_LTF_320M is constructed based on 2xEHT_LTF_160M described in (37), and subcarrier numbers of the sequence 2xEHT_LTF_320M range from –2036 to 2036.

~~[00350]~~[00351] For example, 2xEHT_LTF_320M$_{–2036:2036}$ =

{–2xEHT_LTF_160M$_{–1012:1012}$, 0_{23}, 2xEHT_LTF_160M$_{–1012:1012}$}.

~~[00351]~~[00352] Herein, 2xEHT_LTF_160M$_{–1012:1012}$ in 2xEHT_LTF_320M$_{–2036:2036}$ is

2xEHT_LTF_160M$_{-1012:1012}$ in (37); –2xEHT_LTF_160M$_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_160M$_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[00352]~~[00353] (39) A possible 2x LTF sequence with 240 MHz bandwidth is denoted by 2xEHT_LTF_240M. 2xEHT_LTF_240M is constructed based on 2xEHT_LTF_160M in (37) or based on 2xEHT_LTF_80M in (36). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 2x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 2xEHT_LTF_240M range from –1524 to 1524.

~~[00353]~~[00354] For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{–2xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, –2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}.

~~[00354]~~[00355] Herein, 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ described in (37); 2xEHT_LTF_partA and 2xEHT_LTF_partB are 2xEHT_LTF_partA and 2xEHT_LTF_partB in (36); –2xEHT_LTF_160M$_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_160M$_{-1012:1012}$; –2xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_partA; 0_5 represents five consecutive 0s; and 0_{23} represents 23 consecutive 0s.

~~[00355]~~[00356] For example, 2xEHT_LTF_240M$_{-1524:1524}$ =

{–2xEHT_LTF_80M$_{-500:500}$, 0_{23}, 2xEHT_LTF_160M$_{-1012:1012}$}.

~~[00356]~~[00357] Herein, 2xEHT_LTF_80M$_{-500:500}$ is 2xEHT_LTF_80M$_{-500:500}$ in (36); 2xEHT_LTF_160M$_{-1012:1012}$ is 2xEHT_LTF_160M$_{-1012:1012}$ in (37); –2xEHT_LTF_80M$_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence 2xEHT_LTF_80M$_{-500:500}$; and 0_{23} represents 23 consecutive 0s.

~~[00357]~~[00358] (40) Sequences obtained by performing one or more of the following operations on the 2x LTF sequences with various bandwidth described in (36) to (39), for example, 2xEHT_LTF_80M$_{-500:500}$, 2xEHT_LTF_160M$_{-1012:1012}$, 2xEHT_LTF_240M$_{-1524:1524}$, and 2xEHT_LTF_320M$_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00358]~~[00359] Operation (1): Multiply elements in a sequence by –1.

~~[00359]~~[00360] Operation (2): Reverse an order of elements in a sequence. It is assumed that a sequence is 123456, and a sequence obtained by reversing an order of elements in the sequence is 654321.

~~[00360]~~[00361] Operation (3): Multiply an even-numbered or odd-numbered element in non-zero elements by –1.

~~[00361]~~[00362] As shown in Table 15 below, Table 15 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 2xEHT_LTF_320M described in (38) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00362]~~[00363] For example, an RU26 is used as an example. An 80 MHz sequence includes 36 RU26s, that is, 36 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 36 * 4 = 144 PAPRs may be obtained by considering rotated sequences. A maximum value of the 144 PAPRs is 5.76. 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M are obtained based on the sequence 2xEHT_LTF_80M. PAPR values of 2xEHT_LTF_160M, 2xEHT_LTF_240M, and 2xEHT_LTF_320M on the RU26s are the same as the PAPR values of 2xEHT_LTF_80M on the RU26s.

~~[00363]~~[00364] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00364]~~[00365] It may be learned from Table 15 that values in the second column are all less than values in the third column. That is, maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are all less than corresponding PAPR median values of a BPSK data part. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 2x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on

entire bandwidth.

Table 15: Comparison between PAPR maximum values of 2x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR (dB)	Median PAPR of BPSK Data (dB)
RU26	5.76	5.89
RU52	4.98	6.71
RU52 + RU26	6.23	7.10
RU106	4.55	7.29
RU106 + RU26	5.45	7.44
RU242	5.09	7.94
RU484	5.53	8.44
RU484 + RU242	7.61	8.83
RU996	5.45	8.84
RU996 + RU484	7.71	9.17
RU2*996	5.63 or 8.33	9.27 or 9.28
RU2*996 + RU484	8.37	9.50
RU3*996	7.74	9.54
RU3*996 + RU484	8.19	9.55
RU4*996	8.29	9.60

~~[00365]~~[00366] This embodiment of this application further provides the following LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode.

~~[00366]~~[00367] (41) A possible 4x LTF sequence with 80 MHz bandwidth is denoted by 4xEHT_LTF_80M. Subcarrier numbers of the sequence 4xEHT_LTF_80M range from –500 to 500.

~~[00367]~~[00368] For example, $4xEHT_LTF_80M_{-500:500}$ = {4xEHT_LTF_partA, 0_5 , 4xEHT_LTF_partB}.

~~[00368]~~[00369] Herein, –500 to 500 are subcarrier indexes (the indexes may also be referred to

as numbers), 0_5 represents five consecutive 0s, 4xEHT_LTF_partA includes 498 elements, and 4xEHT_LTF_partB includes 498 elements.

~~[00369]~~[00370] In an example, 4xEHT_LTF_partA =

{–1 –1 –1 1 1 –1 1 1 1 –1 –1 1 1 1
1 1 –1 1 1 –1 –1 –1 –1 –1 1 –1 1 –1
1 1 1 –1 –1 –1 1 –1 1 1 –1 1 –1 –1
1 1 1 –1 1 –1 –1 –1 –1 –1 –1 –1 1 1
–1 –1 –1 1 1 –1 –1 –1 –1 –1 1 1 1 –1
1 –1 1 –1 1 1 –1 1 1 –1 1 –1 –1 –1
–1 –1 1 1 –1 1 –1 1 –1 –1 1 1 –1 –1
–1 –1 –1 1 –1 –1 1 1 –1 1 1 –1 1 –1
–1 1 1 –1 –1 1 –1 –1 –1 1 –1 1 –1 –1
–1 –1 1 1 1 1 1 1 1 –1 1 1 –1 –1
–1 1 –1 –1 1 1 –1 1 –1 1 1 1 –1 –1
1 –1 1 –1 –1 –1 –1 –1 –1 1 –1 –1 1 –1
–1 –1 –1 1 1 1 1 –1 –1 1 –1 –1 –1 –1
1 –1 1 –1 1 –1 –1 1 1 –1 1 1 –1 –1
1 –1 1 –1 –1 1 –1 –1 –1 –1 –1 –1 –1 1
1 1 –1 –1 –1 –1 1 –1 1 –1 –1 –1 1 1
1 1 1 –1 –1 1 1 –1 1 –1 1 1 1 –1
–1 1 1 1 1 1 –1 1 –1 –1 1 1 –1 –1
1 1 1 –1 1 –1 1 1 –1 –1 1 1 1 1
1 1 –1 –1 1 –1 1 –1 –1 –1 –1 1 1 1
–1 –1 –1 –1 –1 –1 –1 1 –1 –1 1 –1 1 –1
–1 1 1 –1 1 1 –1 –1 1 –1 1 –1 1 –1
–1 –1 –1 1 –1 –1 1 1 1 1 –1 –1 –1 –1
1 –1 –1 1 –1 –1 –1 –1 –1 –1 1 –1 1 –1
–1 1 1 1 –1 1 –1 1 1 –1 –1 1 –1 –1
–1 1 1 –1 –1 –1 1 1 –1 1 1 –1 1 –1
–1 1 1 1 –1 1 –1 1 –1 1 –1 –1 –1 –1
–1 –1 –1 1 1 –1 –1 1 1 –1 1 1 1 1

1 –1 –1 1 1 –1 1 –1 1 –1 –1 1 1 1

1 1 –1 1 –1 –1 1 –1 –1 1 –1 1 –1 1

–1 –1 –1 1 1 1 1 1 –1 –1 1 1 1 –1

–1 1 1 1 1 1 1 1 –1 1 –1 –1 –1 1

1 –1 1 –1 –1 1 –1 1 1 1 –1 –1 –1 1

–1 1 –1 1 1 1 1 1 –1 –1 1 –1 –1 –1

–1 –1 1 1 –1 –1 –1 1 –1 –1 1 1 1 –1

1 –1 1 –1 1 1 1 1}

4xEHT_LTF_partB =

{1 1 1 –1 –1 1 –1 –1 –1 1 1 1 –1 –1

1 –1 –1 –1 1 1 –1 –1 –1 –1 –1 1 –1 –1

1 1 1 1 1 –1 1 –1 1 –1 –1 –1 1 1

1 –1 1 –1 –1 1 –1 1 1 –1 –1 –1 1 –1

1 1 1 1 1 1 1 –1 –1 1 1 1 –1 –1

1 1 1 1 1 –1 –1 –1 1 –1 1 –1 1 –1

–1 1 –1 –1 1 –1 1 1 1 1 1 –1 –1 1

–1 1 –1 1 1 –1 –1 1 1 1 1 1 –1 1

1 –1 –1 1 1 –1 –1 –1 –1 –1 –1 –1 1 –1

1 –1 1 –1 1 1 1 –1 –1 1 –1 1 1 –1

1 1 –1 –1 –1 1 1 –1 –1 –1 1 –1 –1 1

1 –1 1 –1 1 1 1 –1 –1 1 –1 1 –1 –1

–1 –1 –1 –1 1 –1 –1 1 –1 –1 –1 –1 1 1

1 1 –1 –1 1 –1 –1 –1 –1 1 –1 1 –1 1

–1 –1 1 1 –1 1 1 –1 –1 1 –1 1 –1 –1

1 –1 –1 –1 –1 –1 –1 –1 1 1 1 1 –1 –1 –1

–1 1 –1 1 –1 –1 1 1 1 1 1 1 –1 –1

1 1 –1 1 –1 1 1 1 –1 –1 1 1 –1 –1

1 1 –1 1 –1 –1 –1 1 1 –1 –1 –1 1 –1

1 –1 –1 1 1 –1 –1 –1 –1 –1 –1 1 1 –1

1 –1 1 1 1 1 –1 –1 –1 1 1 1 1 1

1 1 –1 1 1 –1 1 –1 1 1 –1 –1 1 –1

–1 1 1 –1 1 –1 1 –1 1 1 1 1 –1 1

1 –1 –1 –1 –1 1 1 1 1 –1 1 1 –1 1

1 1 1 1 1 –1 1 –1 1 1 –1 –1 –1 1

–1 1 –1 –1 1 1 –1 1 1 1 –1 –1 1 –1

–1 –1 –1 –1 –1 –1 1 1 1 1 –1 1 –1 1

1 1 –1 1 1 –1 –1 1 1 –1 1 –1 –1 1

–1 –1 1 1 –1 1 1 1 1 1 –1 –1 1 1

–1 1 –1 1 –1 –1 1 1 1 1 1 –1 1 –1

–1 1 –1 –1 1 –1 1 –1 1 –1 –1 –1 1 1

1 1 1 –1 –1 1 1 1 –1 –1 1 1 1 1

1 1 1 –1 1 –1 –1 –1 1 1 –1 1 –1 –1

1 –1 1 1 1 –1 –1 –1 1 –1 1 –1 1 1

1 1 1 –1 –1 1 –1 –1 –1 –1 –1 1 1 –1

–1 –1 1 –1 –1 1 1 1}

~~[00370]~~[00371] (42) A possible 4x LTF sequence with 160 MHz bandwidth is denoted by 4xEHT_LTF_160M. 4xEHT_LTF_160M may be constructed based on 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (41), and subcarrier numbers of the sequence 4xEHT_LTF_160M range from –1012 to 1012.

~~[00371]~~[00372] For example, $4xEHT_LTF_160M_{-1012:1012}$ =

{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, –4xEHT_LTF_partB}.

~~[00372]~~[00373] Herein, 4xEHT_LTF_partA and 4xEHT_LTF_partB are respectively 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (41); –4xEHT_LTF_partB represents negation (multiplied by –1) of all elements in the sequence 4xEHT_LTF_partB; 0_{23} represents 23 consecutive 0s; and 0_5 represents five consecutive 0s.

~~[00373]~~[00374] (43) A possible 4x LTF sequence with 320 MHz bandwidth is denoted by 4xEHT_LTF_320M. 4xEHT_LTF_320M is constructed based on 4xEHT_LTF_160M described in (42) or based on 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB described in (41), and subcarrier numbers of the sequence 4xEHT_LTF_320M range from –2036 to 2036.

~~[00374]~~[00375] For example, $4xEHT_LTF_320M_{-2036:2036}$ =

{$4xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $4xEHT_LTF_80M_{-500:500}$, 0_{23}, –

4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}.

~~[00375]~~[00376] Herein, 4xEHT_LTF_160M$_{-1012:1012}$ is 4xEHT_LTF_160M$_{-1012:1012}$ in (42); 4xEHT_LTF_partA and 4xEHT_LTF_partB are 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (41); –4xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 4xEHT_LTF_partA; 0_{23} represents 23 consecutive 0s; and 0_5 represents five consecutive 0s.

~~[00376]~~[00377] (44) A possible 4x LTF sequence with 240 MHz bandwidth is denoted by 4xEHT_LTF_240M. 4xEHT_LTF_240M is constructed based on 4xEHT_LTF_160M described in (42) and 4xEHT_LTF_80M described in (41). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 4x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 4xEHT_LTF_240M range from –1524 to 1524.

~~[00377]~~[00378] For example, 4xEHT_LTF_240M$_{-1524:1524}$ =

{4xEHT_LTF_160M$_{-1012:1012}$, 0_{23}, 4xEHT_LTF_80M$_{-500:500}$}.

~~[00378]~~[00379] 4xEHT_LTF_80M$_{-500:500}$ is 4xEHT_LTF_80M$_{-500:500}$ in (41); 4xEHT_LTF_160M$_{-1012:1012}$ is 4xEHT_LTF_160M$_{-1012:1012}$ in (42); and 0_{23} represents 23 consecutive 0s.

~~[00379]~~[00380] 4xEHT_LTF_240M$_{-1524:1524}$ =

{4xEHT_LTF_partA, 0_5, –4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_80M$_{-500:500}$, 0_{23}, –4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}.

~~[00380]~~[00381] Herein, 4xEHT_LTF_80M$_{-500:500}$, 4xEHT_LTF_partA, and 4xEHT_LTF_partB are 4xEHT_LTF_80M$_{-500:500}$, 4xEHT_LTF_partA, and 4xEHT_LTF_partB in (41); 0_5 represents five consecutive 0s; 0_{23} represents 23 consecutive 0s; and –4xEHT_LTF_partA and –4xEHT_LTF_partB respectively represent negation (that is, multiplied by –1) of all elements in the sequences 4xEHT_LTF_partA and 4xEHT_LTF_partB.

~~[00381]~~[00382] (45) Sequences obtained by performing one or more of the following operations on the 4x LTF sequences with various bandwidth described in (41) to (44), for example, 4xEHT_LTF_80M$_{-500:500}$, 4xEHT_LTF_160M$_{-1012:1012}$, 4xEHT_LTF_240M$_{-1524:1524}$, and 4xEHT_LTF_320M$_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on

entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00382]~~[00383] Operation (1): Multiply elements in a sequence by –1.

~~[00383]~~[00384] Operation (2): Reverse an order of elements in a sequence.

~~[00384]~~[00385] Operation (3): Multiply an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[00385]~~[00386] As shown in Table 16 below, Table 16 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum PAPR values (a second column) in PAPRs that are of the sequence 4xEHT_LTF_320M described in (43) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00386]~~[00387] For example, an RU52 is used as an example. An 80 MHz sequence includes 16 RU52s, that is, 16 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 16 * 4 = 64 PAPRs may be obtained by considering rotated sequences. A maximum value of the 64 PAPRs is 4.62. 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M are obtained based on the sequence 4xEHT_LTF_80M. PAPR values of 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M on the RU52s are the same as the PAPR values of 4xEHT_LTF_80M on the RU52s.

~~[00387]~~[00388] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00388]~~[00389] It may be learned from Table 16 that values in the second column are all less than values in the third column. That is, most of maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are less than corresponding PAPR median values of a BPSK data part. In addition, PAPRs are particularly low for important RUs such as RU4*996, RU2*996, and RU996. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on

a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 16: Comparison between PAPR maximum values of 4x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR	Median PAPR of BPSK Data
RU26	4.17	5.89
RU52	4.62	6.71
RU52 + RU26	5.37	7.10
RU106	4.98	7.29
RU106 + RU26	5.72	7.44
RU242	5.45	7.94
RU484	5.79	8.44
RU484 + RU242	7.95	8.83
RU996	5.80	8.84
RU996 + RU484	8.00	9.17
RU2*996	6.03	9.27 or 9.28
RU2*996 + RU484	9.25	9.50
RU3*996	8.02	9.54
RU3*996 + RU484	8.17	9.55
RU4*996	6.07	9.60

~~[00389]~~[00390] (46) A possible 4x LTF sequence with 80 MHz bandwidth is denoted by 4xEHT_LTF_80M. Subcarrier numbers of the sequence 4xEHT_LTF_80M range from –500 to 500.

~~[00390]~~[00391] For example, $4xEHT_LTF_80M_{-500:500}$ = {4xEHT_LTF_partA, 0_5 , 4xEHT_LTF_partB}.

~~[00391]~~[00392] Herein, –500 to 500 are subcarrier indexes (the indexes may also be referred to as numbers), 0_5 represents five consecutive 0s, 4xEHT_LTF_partA includes 498 elements, and 4xEHT_LTF_partB includes 498 elements.

~~[00392]~~[00393] In an example, 4xEHT_LTF_partA =

{1 –1 –1 –1 –1 1 –1 –1 1 –1 –1 –1 1 1

–1 –1 –1 1 –1 –1 1 –1 1 1 1 –1 1 –1

1 –1 –1 –1 –1 1 1 1 1 1 –1 –1 1 –1

1 –1 –1 –1 1 1 –1 –1 1 –1 –1 –1 1 1

1 –1 –1 1 1 –1 –1 1 –1 1 1 –1 1 –1

1 1 1 –1 1 –1 1 1 1 1 1 1 –1 –1

–1 1 –1 1 –1 –1 –1 1 –1 –1 1 1 1 1

1 1 –1 1 –1 1 1 –1 1 –1 1 –1 1 –1

1 –1 –1 1 1 1 1 –1 –1 –1 –1 –1 –1 –1

–1 1 –1 –1 1 –1 –1 1 1 1 –1 1 –1 –1

–1 1 1 1 –1 1 1 –1 –1 1 –1 –1 –1 1

1 1 1 –1 1 1 1 1 1 1 –1 1 –1 –1

1 –1 1 –1 –1 1 1 1 1 1 –1 1 1 –1

–1 1 1 1 –1 1 1 –1 1 1 –1 –1 1 1

–1 –1 –1 –1 1 1 1 1 1 –1 1 1 1 1

1 –1 1 –1 1 –1 –1 1 –1 –1 –1 –1 –1 1

–1 –1 –1 1 1 –1 1 –1 1 –1 –1 –1 –1 –1

1 1 1 1 –1 1 –1 –1 1 1 –1 –1 –1 1

1 1 1 1 –1 1 –1 1 –1 –1 1 1 1 –1

1 1 1 1 1 –1 1 1 –1 1 –1 1 –1 –1

–1 –1 –1 1 –1 –1 –1 –1 –1 1 1 1 1 –1

–1 1 1 –1 –1 1 –1 –1 1 –1 –1 –1 1 1

–1 –1 1 –1 –1 1 –1 1 1 –1 –1 –1 1 1

–1 –1 –1 –1 1 –1 1 1 –1 1 –1 1 1 1

1 1 1 –1 –1 1 –1 –1 –1 1 –1 1 –1 –1

–1 1 1 1 1 1 1 –1 1 –1 1 1 1 –1

1 –1 1 1 –1 1 –1 –1 1 1 –1 –1 1 1

1 –1 –1 –1 1 –1 –1 1 1 –1 –1 –1 1 –1

1 –1 –1 1 1 1 1 1 –1 –1 –1 –1 1 –1

1 –1 1 1 1 –1 1 –1 –1 1 –1 –1 –1 1

1 –1 –1 –1 1 –1 –1 1 –1 –1 –1 –1 1 –1

1 1 –1 –1 –1 1 –1 –1}

4xEHT_LTF_partB =

{–1 –1 1 –1 1 1 1 1 1 1 –1 –1 –1 –1

1 –1 –1 1 –1 –1 –1 1 1 –1 –1 –1 1 –1

–1 1 –1 1 1 1 –1 1 –1 1 –1 –1 –1 –1

1 1 1 1 1 –1 –1 1 –1 1 –1 –1 –1 1

1 –1 –1 1 –1 –1 –1 1 1 1 –1 –1 1 1

–1 –1 1 –1 1 1 –1 1 –1 1 1 1 –1 1

–1 1 1 1 1 1 1 –1 –1 –1 1 –1 1 –1

–1 –1 1 –1 –1 1 1 1 1 1 1 –1 1 –1

1 1 –1 1 –1 –1 –1 –1 1 1 –1 –1 –1 1

1 –1 1 –1 –1 1 –1 –1 –1 1 –1 1 –1 1

–1 –1 –1 1 –1 1 –1 1 1 1 –1 –1 –1 1

–1 –1 1 1 –1 1 1 1 –1 –1 –1 –1 1 –1

–1 –1 –1 –1 –1 1 –1 1 1 –1 1 –1 1 1

–1 –1 –1 –1 –1 1 –1 –1 1 1 –1 –1 –1 1

–1 –1 1 –1 –1 1 1 –1 –1 1 1 1 1 –1

–1 –1 1 –1 –1 1 1 1 1 1 –1 –1 1 1

–1 –1 –1 –1 –1 –1 –1 –1 1 1 1 1 1 –1

1 –1 1 –1 –1 1 1 1 –1 1 1 1 1 1

–1 1 1 –1 1 –1 1 –1 –1 –1 –1 –1 1 –1

–1 –1 –1 –1 1 1 1 1 –1 –1 1 1 –1 –1

1 –1 –1 1 –1 –1 –1 1 1 –1 –1 1 –1 –1

–1 –1 –1 1 1 –1 1 –1 1 1 –1 1 –1 –1

–1 –1 –1 –1 1 –1 –1 –1 –1 1 1 1 –1 1

1 –1 –1 1 –1 –1 –1 1 1 1 –1 1 –1 –1

–1 1 1 –1 1 1 –1 1 1 1 1 1 1 1

1 –1 –1 –1 –1 1 1 –1 1 –1 1 –1 1 –1

1 –1 –1 1 –1 1 –1 –1 –1 –1 –1 –1 1 1

–1 1 1 1 –1 1 –1 1 1 1 –1 –1 –1 –1

–1 –1 1 –1 1 –1 –1 –1 1 –1 1 –1 –1 1

–1 1 1 –1 –1 1 1 –1 –1 –1 1 1 1 –1

1 1 –1 –1 1 1 1 –1 1 –1 1 1 –1 –1

–1 –1 –1 1 1 1 1 –1 1 –1 1 –1 –1 –1

1 –1 1 1 –1 1 1 1 –1 –1 1 1 1 –1

1 1 –1 1 1 1 1 –1}

~~[00393]~~[00394] (47) A possible 4x LTF sequence with 160 MHz bandwidth is denoted by 4xEHT_LTF_160M. 4xEHT_LTF_160M may be constructed based on 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (46), and subcarrier numbers of the sequence 4xEHT_LTF_160M range from –1012 to 1012.

~~[00394]~~[00395] For example, $4xEHT_LTF_160M_{-1012:1012}$ =

{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, –4xEHT_LTF_partB}.

~~[00395]~~[00396] Herein, 4xEHT_LTF_partA and 4xEHT_LTF_partB are respectively 4xEHT_LTF_partA and 4xEHT_LTF_partB described in (46); –4xEHT_LTF_partB represents negation (multiplied by –1) of all elements in the sequence 4xEHT_LTF_partB; 0_{23} represents 23 consecutive 0s; and 0_5 represents five consecutive 0s.

~~[00396]~~[00397] (48) A possible 4x LTF sequence with 320 MHz bandwidth is denoted by 4xEHT_LTF_320M. 4xEHT_LTF_320M is constructed based on 4xEHT_LTF_160M described in (47), and subcarrier numbers of the sequence 4xEHT_LTF_320M range from –2036 to 2036.

~~[00397]~~[00398] For example, $4xEHT_LTF_320M_{-2036:2036}$ =

{$4xEHT_LTF_160M_{-1012:1012}$, 0_{23}, $-4xEHT_LTF_160M_{-1012:1012}$}.

~~[00398]~~[00399] Herein, $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (47); $-4xEHT_LTF_160M_{-1012:1012}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_160M_{-1012:1012}$; and 0_{23} represents 23 consecutive 0s.

~~[00399]~~[00400] (49) A possible 4x LTF sequence with 240 MHz bandwidth is denoted by

4xEHT_LTF_240M. 4xEHT_LTF_240M is constructed based on 4xEHT_LTF_160M described in (47) and 4xEHT_LTF_80M described in (46), or based on –4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB described in (46). For example, when an 80 MHz channel in a 320 MHz channel is missing, a 240 MHz channel is formed, and the formed 240 MHz channel may be continuous or discontinuous in frequency domain. A punctured 4x EHT LTF sequence corresponding to 320 MHz bandwidth may be used as a sequence with 240 MHz bandwidth. Subcarrier numbers of the sequence 4xEHT_LTF_240M range from –1524 to 1524.

~~[00400]~~[00401] For example, $4xEHT_LTF_240M_{-1524:1524}$ =

$$\{4xEHT_LTF_160M_{-1012:1012},\ 0_{23},\ -4xEHT_LTF_80M_{-500:500}\}.$$

~~[00401]~~[00402] Herein, $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (47); $4xEHT_LTF_80M_{-500:500}$ is $4xEHT_LTF_80M_{-500:500}$ in (46); $-4xEHT_LTF_80M_{-500:500}$ represents negation (that is, multiplied by –1) of all elements in the sequence $4xEHT_LTF_80M_{-500:500}$.

~~[00402]~~[00403] Alternatively, $4xEHT_LTF_240M_{-1524:1524}$ =

$$\{-4xEHT_LTF_partA,\ 0_5,\ 4xEHT_LTF_partB,\ 0_{23},\ 4xEHT_LTF_160M_{-1012:1012}\}.$$

~~[00403]~~[00404] Herein, 4xEHT_LTF_partA and 4xEHT_LTF_partB are 4xEHT_LTF_partA and 4xEHT_LTF_partB in (46); $4xEHT_LTF_160M_{-1012:1012}$ is $4xEHT_LTF_160M_{-1012:1012}$ in (47); –4xEHT_LTF_partA represents negation (that is, multiplied by –1) of all elements in the sequence 4xEHT_LTF_partA; 0_{23} represents 23 consecutive 0s; and 0_5 represents five consecutive 0s.

~~[00404]~~[00405] (50) Sequences obtained by performing one or more of the following operations on the 4x LTF sequences with various bandwidth described in (46) to (49), for example, $4xEHT_LTF_80M_{-500:500}$, $4xEHT_LTF_160M_{-1012:1012}$, $4xEHT_LTF_240M_{-1524:1524}$, and $4xEHT_LTF_320M_{-2036:2036}$, are also sequences to be protected in this application. After the following operations, PAPR values of these sequences on a single RU, on a combined RU, on entire bandwidth, and in a considered multi-stream scenario do not change.

~~[00405]~~[00406] Operation (1): Multiply elements in a sequence by –1.

~~[00406]~~[00407] Operation (2): Reverse an order of elements in a sequence.

~~[00407]~~[00408] Operation (3): Multiply an element at an even-numbered or odd-numbered position in a sequence by –1.

~~[00408]~~[00409] As shown in Table 17 below, Table 17 provides a comparison result between PAPR median values (a third column) of a BPSK (a modulation mode) data part and maximum

PAPR values (a second column) in PAPRs that are of the sequence 4xEHT_LTF_320M described in (48) and a plurality of corresponding rotated sequences and that are on different single-RUs, various combined RUs, and entire bandwidth.

~~[00409]~~[00410] For example, an RU52 is used as an example. An 80 MHz sequence includes 16 RU52s, that is, 16 PAPRs. One sequence may be rotated to obtain four sequences. Therefore, 16 * 4 = 64 PAPRs may be obtained by considering rotated sequences. A maximum value of the 64 PAPRs is 4.64. 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M are obtained based on the sequence 4xEHT_LTF_80M. PAPR values of 4xEHT_LTF_160M, 4xEHT_LTF_240M, and 4xEHT_LTF_320M on the RU52s are the same as the PAPR values of 4xEHT_LTF_80M on the RU52s.

~~[00410]~~[00411] It should be noted that there are two maximum PAPR values corresponding to an RU2*996. This is because a 160 MHz channel may include two continuous 80 MHz channels, or may include two discontinuous 80 MHz channels. Therefore, two maximum PAPR values are obtained, and two PAPR median values of a data part are also obtained.

~~[00411]~~[00412] It may be learned from Table 17 that most of values in the second column are less than values in the third column. That is, most of maximum PAPR values on different single-RUs, various combined RUs, and entire bandwidth under considered influence of phase rotation are less than corresponding PAPR median values of a BPSK data part. In addition, PAPRs are particularly low for important RUs such as RU4*996, RU2*996, and RU996. Therefore, it is verified that the LTF sequences with 80 MHz bandwidth, 160 MHz bandwidth, 240 MHz bandwidth, and 320 MHz bandwidth in the 4x mode that are generated in this application have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth. In addition, a multi-stream scenario is also considered, and rotated sequences obtained after phase rotation is performed on these sequences have relatively low PAPR values on a single RU, relatively low PAPR values on a combined RU, and relatively low PAPR values on entire bandwidth.

Table 17: Comparison between PAPR maximum values of 4x EHT LTF sequences on a single RU, a combined RU, and entire bandwidth under considered influence of a plurality of P-matrices and PAPR median values of a BPSK data part

RU Size	Max PAPR	Median PAPR of BPSK Data
RU26	4.20	5.89
RU52	4.64	6.71
RU52 + RU26	4.90	7.10
RU106	5.42	7.29
RU106 + RU26	5.76	7.44
RU242	5.42	7.94
RU484	5.71	8.44
RU484 + RU242	7.85	8.83
RU996	5.89	8.84
RU996 + RU484	7.90	9.17
RU2*996	6.17	9.27 or 9.28
RU2*996 + RU484	8.47	9.50
RU3*996	8.15	9.54
RU3*996 + RU484	8.34	9.55
RU4*996	8.44	9.60

~~[00412]~~[00413] The foregoing describes the method for transmitting a PPDU in the embodiments of this application. The following describes an apparatus for transmitting a PPDU in the embodiments of this application. The apparatus for transmitting a PPDU in the embodiments of this application includes an apparatus for transmitting a PPDU and applied to a transmit end and an apparatus for transmitting a PPDU and applied to a receive end. It should be understood that the apparatus for transmitting a PPDU and applied to a transmit end is the first communications device in the foregoing method, and has any function of the first communications device in the foregoing method; and the apparatus for transmitting a physical layer protocol data unit and applied to a receive end is the second communications device in the foregoing method, and has any function of the second communications device in the foregoing method.

~~[00413]~~[00414] In the embodiments of this application, the communications device may be

divided into functional units based on the foregoing method example. For example, each functional unit may be obtained through division based on each corresponding function, or two or more functions may be integrated into one processing unit. The integrated unit may be implemented in a form of hardware, or may be implemented in a form of a software functional unit. It should be noted that in the embodiments of this application, division into the units is an example and is merely logical function division, and may be other division in an actual implementation.

~~[00414]~~[00415] FIG. 9 is a schematic diagram of a structure of an apparatus for transmitting a PPDU and applied to a transmit end according to an embodiment of this application. The apparatus includes a processing unit 11 and a transceiver unit 12.

~~[00415]~~[00416] The processing unit 11 is configured to generate a physical layer protocol data unit PPDU, where the PPDU includes an LTF sequence.

~~[00416]~~[00417] The transceiver unit 12 is configured to send the PPDU.

~~[00417]~~[00418] Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00418]~~[00419] The apparatus for transmitting a PPDU, applied to a transmit end, and provided in this embodiment of this application is the first communications device in the foregoing method, and has any function of the first communications device in the foregoing method. For specific details, refer to the foregoing method. Details are not described herein.

~~[00419]~~[00420] FIG. 10 is a schematic diagram of a structure of an apparatus for transmitting a PPDU and applied to a receive end according to an embodiment of this application. The apparatus includes a transceiver unit 21 and a processing unit 22.

~~[00420]~~[00421] The transceiver unit 21 is configured to receive a PPDU, where the PPDU includes an LTF sequence.

~~[00421]~~[00422] The processing unit 22 is configured to parse the PPDU to obtain the LTF sequence.

~~[00422]~~[00423] Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00423]~~[00424] The apparatus for transmitting a PPDU, applied to a receive end, and provided in this embodiment of this application is the second communications device in the foregoing method, and has any function of the second communications device in the foregoing method. For specific details, refer to the foregoing method. Details are not described herein.

~~[00424]~~[00425] The foregoing describes the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end in the embodiments of this application. The following describes possible product forms of the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end. It should be understood that any form of product having the features of the apparatus for transmitting a PPDU and applied to a transmit end in FIG. 9 and any form of product having the features of the apparatus for transmitting a PPDU and applied to a receive end in FIG. 10 fall within the protection scope of this application. It should be further understood that the following description is merely an example, and does not limit a product form of the apparatus for transmitting a PPDU and applied to a transmit end and a product form of the apparatus for transmitting a physical layer protocol data unit and applied to a receive end in the embodiments of this application.

~~[00425]~~[00426] In a possible product form, the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end in the embodiments of this application may be implemented by using a general bus architecture.

~~[00426]~~[00427] The apparatus for transmitting a PPDU and applied to a transmit end includes a processor and a transceiver that is internally connected to the processor for communication. The processor is configured to generate a PPDU, where the PPDU includes an LTF sequence. The transceiver is configured to send the PPDU. Optionally, the apparatus for transmitting a PPDU and applied to a transmit end may further include a memory. The memory is configured to store instructions executed by the processor. Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00427]~~[00428] The apparatus for transmitting a PPDU and applied to a receive end includes a processor and a transceiver that is internally connected to the processor for communication. The transceiver is configured to receive a PPDU. The processor is configured to parse the received PPDU to obtain an LTF sequence included in the PPDU. Optionally, the apparatus for transmitting a PPDU and applied to a receive end may further include a memory. The memory is configured to store instructions executed by the processor. Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00428]~~[00429] In a possible product form, the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end in the

embodiments of this application may be implemented by using a general-purpose processor.

~~[00429]~~[00430] A general-purpose processor that implements the apparatus for transmitting a PPDU and applied to a transmit end includes a processing circuit and an input/output interface that is internally connected to the processing circuit for communication. The processing circuit is configured to generate a PPDU, where the PPDU includes an LTF sequence. The input/output interface is configured to send the PPDU. Optionally, the general-purpose processor may further include a storage medium. The storage medium is configured to store instructions executed by the processing circuit. Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00430]~~[00431] A general-purpose processor that implements the apparatus for transmitting a PPDU and applied to a receive end includes a processing circuit and an input/output interface that is internally connected to the processing circuit for communication. The input/output interface is configured to receive a PPDU, where the PPDU includes an LTF sequence. The processing circuit is configured to parse the PPDU PPDU to obtain the LTF sequence included in the PPDU. Optionally, the general-purpose processor may further include a storage medium. The storage medium is configured to store instructions executed by the processing circuit. Optionally, the LTF sequence included in the PPDU may be any LTF sequence provided in (1) to (50).

~~[00431]~~[00432] In a possible product form, the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end in the embodiments of this application may alternatively be implemented by using the following components: one or more FPGAs (field programmable gate arrays), PLDs (programmable logic devices), controllers, state machines, gate logic, discrete hardware components, any other suitable circuits, or any combination of circuits that can perform various functions described in this application.

~~[00432]~~[00433] It should be understood that the apparatus for transmitting a PPDU and applied to a transmit end and the apparatus for transmitting a PPDU and applied to a receive end in the foregoing product forms respectively have any functions of the first communications device and the second communications device in the foregoing method embodiment. Details are not described herein.

~~[00433]~~[00434] An embodiment of this application further provides a computer-readable storage medium. The computer-readable storage medium stores instructions, and when the instructions are

run on a computer, the computer is enabled to perform the foregoing method for transmitting a PPDU.

~~[00434]~~[00435] An embodiment of this application further provides a computer program product. When the computer program product is run on a computer, the computer is enabled to perform the foregoing method for transmitting a PPDU.

~~[00435]~~[00436] An embodiment of this application further provides a wireless communications system, including a first communications device (for example, an AP) and a second communications device (for example, a STA). The first communications device and the second communications device may perform the foregoing method for transmitting a PPDU.

~~[00436]~~[00437] A person of ordinary skill in the art may be aware that, in combination with the examples described in the embodiments disclosed in this specification, method steps and units may be implemented by electronic hardware, computer software, or a combination thereof. To clearly describe the interchangeability between the hardware and the software, the foregoing has generally described steps and compositions of each embodiment based on functions. Whether the functions are performed by hardware or software depends on particular applications and design constraints of the technical solutions. A person of ordinary skill in the art may use different methods to implement the described functions for each particular application, but it should not be considered that the implementation goes beyond the scope of this application.

~~[00437]~~[00438] It can be clearly understood by a person skilled in the art that, for the purpose of convenient and brief description, for a detailed working process of the foregoing described system, apparatus, and unit, refer to a corresponding process in the foregoing method embodiment. Details are not described herein again.

~~[00438]~~[00439] In the several embodiments provided in this application, it should be understood that the disclosed system, apparatus, and method may be implemented in other manners. For example, the described apparatus embodiments are merely examples. For example, division into units is merely logical function division and may be other division during actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented through some interfaces, indirect couplings or communication connections between the apparatuses or units, or electrical connections, mechanical connections, or connections in other forms.

~~[00439]~~**[00440]** The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may be distributed on a plurality of network units. Some or all of the units may be selected based on actual requirements to achieve the objectives of the solutions of the embodiments of this application.

~~[00440]~~**[00441]** In addition, functional units in the embodiments of this application may be integrated into one processing unit, or each of the units may exist alone physically, or two or more units are integrated into one unit. The integrated unit may be implemented in a form of hardware, or may be implemented in a form of a software functional unit.

~~[00441]~~**[00442]** When the integrated unit is implemented in the form of a software functional unit and is sold or used as an independent product, the integrated unit may be stored in a computer-readable storage medium. Based on such an understanding, the technical solutions in this application essentially, or the part contributing to the prior art, or all or some of the technical solutions may be implemented in the form of a software product. The computer software product is stored in a storage medium and includes several instructions for instructing a computer device (which may be a personal computer, a server, a network device, or the like) to perform all or some of the steps of the methods described in the embodiments of this application. The foregoing storage medium includes: any medium that can store program code, such as a USB flash drive, a removable hard disk, a read-only memory (read-only memory, ROM), a random access memory (random access memory, RAM), a magnetic disk, or an optical disc.

~~[00442]~~**[00443]** It should be noted that the term "and/or" describes an association relationship for describing associated objects and represents that three relationships may exist. For example, A and/or B may represent the following three cases: Only A exists, both A and B exist, and only B exists. The character "/" generally indicates an "or" relationship between the associated objects. "At least one" means one or more. Similar to "A and/or B", "at least one of A and B" describes an association relationship for describing associated objects and represents that three relationships may exist. For example, at least one of A and B may represent the following three cases: Only A exists, both A and B exist, and only B exists.

~~[00443]~~**[00444]** A person skilled in the art should understand that the embodiments of this application may be provided as a method, a system, or a computer program product. Therefore, this application may use a form of hardware only embodiments, software only embodiments, or

embodiments with a combination of software and hardware. Moreover, this application may use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, an optical memory, and the like) that include computer usable program code.

~~[00444]~~[00445] This application is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of this application. It should be understood that computer program instructions may be used to implement each process and/or each block in the flowcharts and/or the block diagrams and a combination of a process and/or a block in the flowcharts and/or the block diagrams. These computer program instructions may be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of another programmable data processing device to generate a machine, so that the instructions executed by the computer or the processor of the another programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

~~[00445]~~[00446] These computer program instructions may alternatively be stored in a computer-readable memory that can instruct the computer or the another programmable data processing device to work in a specific manner, so that the instructions stored in the computer-readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

~~[00446]~~[00447] These computer program instructions may alternatively be loaded onto the computer or the another programmable data processing device, so that a series of operations and steps are performed on the computer or the another programmable device, to generate computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

~~[00447]~~[00448] Although some embodiments of this application have been described, persons skilled in the art can make changes and modifications to these embodiments once they learn the basic inventive concept. Therefore, the following claims are intended to be construed as to cover the embodiments and all changes and modifications falling within the scope of this application.

~~[00448]~~**[00449]** Clearly, persons skilled in the art can make various modifications and variations to the embodiments of this application without departing from the spirit and scope of the embodiments of this application. This application is also intended to cover these modifications and variations to embodiments of this application provided that the modifications and variations fall within the scope of protection defined by the following claims and their equivalent technologies.

CLAIMS

~~What is claimed is:~~

1. A method for transmitting a physical layer protocol data unit, comprising:

generating, ~~by a first communications device,~~ a physical layer protocol data unit, PPDU, wherein the PPDU comprises a long training field carrying an LTF sequence~~; and~~, ~~sending, by the first communications device, the PPDU.~~ a 320 MHz 4x LTF sequence is $4xEHT_LTF_320M_{-2036:2036}$, $4xEHT_LTF_320M_{-2036:2036}$ $=\{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, -4xEHT_LTF_partB, 0_{23}, -4xEHT_LTF_partA, 0_5, -4xEHT_LTF_partB, 0_{23}, -4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB\}$, wherein $4xEHT_LTF_partA$ is a sequence segment including ± 1, $4xEHT_LTF_partB$ is a sequence segment including ± 1, 0_5 represents 5 consecutive 0s, 0_{23} represents 23 consecutive 0s;

sending the PPDU.

2. The method according to claim 1, $4xEHT_LTF_partA$ includes 498 elements, $4xEHT_LTF_partB$ includes 498 elements.

3. The method according to claim 1 or 2, wherein

$4xEHT_LTF_partA =$

$\{1\ -1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$
$-1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ 1\ 1\ 1\ -1\ 1\ -1$
$1\ -1\ -1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ 1\ -1$
$1\ -1\ -1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$
$1\ -1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ 1\ 1\ -1\ 1\ -1$
$1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ -1$
$-1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ 1$
$1\ 1\ -1\ 1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1$
$1\ -1\ -1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ -1$
$-1\ 1\ -1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ -1\ 1\ -1\ -1$
$-1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1$
$1\ 1\ 1\ -1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ -1\ -1$
$1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ -1$
$-1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ 1\ 1\ -1\ -1\ 1\ 1$
$-1\ -1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ 1\ 1$
$1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ -1\ -1\ 1$
$-1\ -1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1\ -1\ -1\ -1$
$1\ 1\ 1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ -1\ -1\ -1\ 1$
$1\ 1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ -1$
$1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1$

```
    -1   -1   -1    1   -1   -1   -1   -1   -1    1    1    1    1   -1
    -1    1    1   -1   -1    1   -1   -1    1   -1   -1   -1    1    1
    -1   -1    1   -1   -1   -1   -1   -1    1    1   -1    1   -1    1
     1   -1    1   -1   -1   -1   -1   -1   -1    1   -1   -1   -1   -1
     1    1    1   -1    1    1   -1   -1    1   -1   -1   -1    1    1
     1   -1    1   -1    1   -1   -1   -1    1   -1    1   -1    1   -1
    -1   -1    1   -1   -1    1   -1    1    1   -1   -1   -1    1    1
    -1   -1   -1   -1    1   -1    1    1   -1    1   -1    1    1    1
     1    1    1   -1   -1    1   -1   -1   -1    1   -1    1   -1   -1
    -1    1    1    1    1    1    1   -1    1   -1    1    1    1   -1
     1   -1    1    1   -1    1   -1   -1    1    1   -1   -1    1    1
     1   -1   -1   -1    1   -1   -1    1    1   -1   -1   -1    1   -1
     1   -1   -1    1    1    1    1    1   -1   -1   -1   -1    1   -1
     1   -1    1    1    1   -1    1   -1   -1    1   -1   -1   -1    1
     1   -1   -1   -1    1   -1   -1    1   -1   -1   -1   -1    1   -1
     1    1   -1   -1   -1    1   -1   -1};

    4xEHT_LTF_partB =

{-1   -1    1   -1    1    1    1    1    1    1   -1   -1   -1   -1
     1   -1   -1    1   -1   -1   -1    1    1   -1   -1   -1    1   -1
    -1    1   -1    1    1    1   -1    1   -1    1   -1   -1   -1   -1
     1    1    1    1    1   -1   -1    1   -1    1   -1   -1   -1    1
     1   -1   -1    1   -1   -1   -1    1    1    1   -1   -1    1    1
    -1   -1    1   -1    1    1   -1    1   -1    1    1    1   -1    1
    -1    1    1    1    1    1    1   -1   -1   -1    1   -1    1   -1
    -1   -1    1   -1   -1    1    1    1    1    1    1   -1    1   -1
     1    1   -1    1   -1   -1   -1   -1    1    1   -1   -1   -1    1
     1   -1    1   -1   -1    1   -1   -1   -1    1   -1    1   -1    1
    -1   -1   -1    1   -1    1   -1    1    1    1   -1   -1   -1    1
    -1   -1    1    1   -1    1    1    1   -1   -1   -1   -1    1   -1
    -1   -1   -1   -1   -1    1   -1    1    1   -1    1   -1    1    1
    -1   -1    1   -1   -1    1    1   -1   -1    1    1    1    1   -1
    -1   -1   -1   -1    1   -1   -1   -1   -1   -1    1   -1    1   -1
     1    1   -1    1    1    1    1    1   -1    1    1    1   -1   -1
     1   -1    1   -1    1    1    1    1    1   -1   -1   -1    1    1
    -1   -1   -1   -1   -1   -1   -1   -1    1    1    1    1    1   -1
     1   -1    1   -1   -1    1    1    1   -1    1    1    1    1    1
    -1    1    1   -1    1   -1    1   -1   -1   -1   -1   -1    1   -1
    -1   -1   -1   -1    1    1    1    1   -1   -1    1    1   -1   -1
     1   -1   -1    1   -1   -1   -1    1    1   -1   -1    1   -1   -1
```

−1 −1 −1 1 1 −1 1 −1 1 1 −1 1 −1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 1 1 1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1

1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1

−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1

−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1

−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1

1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1

−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1

1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1

1 1 −1 1 1 1 1 −1 }.

42. A method for transmitting a physical layer protocol data unit, comprising:

receiving~~, by a second communications device,~~ a physical layer protocol data unit, PPDU; and

parsing~~, by the second communications device,~~ the received PPDU to obtain ~~a long training field~~ an LTF sequence carried in a long training field ~~comprised in~~of the PPDU~~.~~;and

estimating a channel according to the obtained LTF sequence and a specified LTF sequence, wherein a specified 320 MHz 4x LTF sequence is $4xEHT_LTF_320M_{-2036:2036}$,

$4xEHT_LTF_320M_{-2036:2036}=\{4xEHT_LTF_160M_{-1012:1012}, 0_{23}, -4xEHT_LTF_160M_{-1012:1012}\}=$

$\{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, -4xEHT_LTF_partB, 0_{23}, -4xEHT_LTF_partA, 0_5, -4xEHT_LTF_partB, 0_{23}, -4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB\}$, wherein $4xEHT_LTF_partA$ is a sequence segment including ± 1, $4xEHT_LTF_partB$ is a sequence segment including ± 1, 0_5 represents 5 consecutive 0s, 0_{23} represents 23 consecutive 0s.

5. The method according to claim 4, $4xEHT_LTF_partA$ includes 498 elements, $4xEHT_LTF_partB$ includes 498 elements.

6. The method according to claim 4 or 5, wherein

$4xEHT_LTF_partA =$

$\{$ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1

1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1

1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

1 1 1 −1 1 −1 1 1 1 1 1 1 −1 −1

−1 1 −1 1 −1 −1 −1 1 −1 −1 1 1 1 1

1 1 −1 1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 1 1 1 −1 −1 −1 −1 −1 −1 −1

−1 1 −1 −1 1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1

1 1 1 −1 1 1 1 1 1 1 −1 1 −1 −1

1 −1 1 −1 −1 1 1 1 1 1 −1 1 1 −1

−1 1 1 1 −1 1 1 −1 1 1 −1 −1 1 1

−1 −1 −1 −1 1 1 1 1 1 −1 1 1 1 1

1 −1 1 −1 1 −1 −1 1 −1 −1 −1 −1 −1 1

−1 −1 −1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1

1 1 1 1 −1 1 −1 −1 1 1 −1 −1 −1 1

1 1 1 1 −1 1 −1 1 −1 −1 1 1 1 −1

1 1 1 1 1 −1 1 1 −1 1 −1 1 −1 −1

−1 −1 −1 1 −1 −1 −1 −1 −1 1 1 1 1 −1

−1 1 1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 1 −1 −1 −1 −1 −1 1 1 −1 1 −1 1

1 −1 1 −1 −1 −1 −1 −1 −1 1 −1 −1 −1 −1

1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 −1 1 1 −1 −1 −1 1 1

1 1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 −1

−1 1 1 1 1 1 1 −1 1 −1 1 1 1 −1

1 −1 1 1 −1 1 −1 −1 1 1 −1 −1 1 1

1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1 −1

1 −1 −1 1 1 1 1 1 −1 −1 −1 −1 1 −1

1 −1 1 1 1 −1 1 −1 −1 1 −1 −1 −1 1

1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 −1 1 −1

1 1 −1 −1 −1 1 −1 −1 };

4xEHT_LTF_partB =

{−1 −1 1 −1 1 1 1 1 1 1 −1 −1 −1 −1

1 −1 −1 1 −1 −1 −1 1 1 −1 −1 −1 1 −1

−1 1 −1 1 1 1 −1 1 −1 1 −1 −1 −1 −1

1 1 1 1 1 −1 −1 1 −1 1 −1 −1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 −1 1 1

−1 −1 1 −1 1 1 −1 1 −1 1 1 1 −1 1

−1 1 1 1 1 1 1 −1 −1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 1 1 1 1 1 −1 1 −1

1 1 −1 1 −1 −1 −1 −1 1 1 −1 −1 −1 1

1 −1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 1

−1 −1 −1 1 −1 1 −1 1 1 1 −1 −1 −1 1

−1 −1 1 1 −1 1 1 1 −1 −1 −1 −1 1 −1

−1 −1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1

−1 −1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1

−1 −1 1 −1 −1 1 1 −1 −1 1 1 1 1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 −1 1 −1 1 −1

1 1 −1 1 1 1 1 1 −1 1 1 1 −1 −1

1 −1 1 −1 1 1 1 1 1 −1 −1 −1 1 1

−1 −1 −1 −1 −1 −1 −1 −1 1 1 1 1 1 −1

1 −1 1 −1 −1 1 1 1 −1 1 1 1 1 1

−1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1 1 −1

−1 −1 −1 −1 1 1 1 1 −1 −1 1 1 −1 −1

1 −1 −1 1 −1 −1 −1 1 1 −1 −1 1 −1 −1

−1 −1 −1 1 1 −1 1 −1 1 1 −1 1 −1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 1 1 1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1

1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1

−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1

−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1

−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1

1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1

−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1

1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1

1 1 −1 1 1 1 1 −1}.

7. An apparatus for transmitting a physical layer protocol data unit, comprising:

a generating unit, configured to generate a physical layer protocol data unit, PPDU, wherein the PPDU comprises a long training field carrying an LTF sequence, a 320 MHz 4x LTF sequence is $4xEHT_LTF_320M_{-2036:2036}$, $4xEHT_LTF_320M_{-2036:2036}$ = {4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB }, wherein 4xEHT_LTF_partA is a sequence segment including ±1, 4xEHT_LTF_partB is a sequence segment including ±1, 0_5 represents 5 consecutive 0s, 0_{23} represents 23 consecutive 0s;

a sending unit, configured to send the PPDU.

8. The apparatus according to claim 7, 4xEHT_LTF_partA includes 498 elements, 4xEHT_LTF_partB includes 498 elements.

9. The apparatus according to claim 7 or 8, wherein

4xEHT_LTF_partA =

$$\{$$
$$1\ -1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$$
$$-1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ 1\ 1\ 1\ -1\ 1\ -1$$
$$1\ -1\ -1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ 1\ -1$$
$$1\ -1\ -1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$$
$$1\ -1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ 1\ 1\ -1\ 1\ -1$$
$$1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ -1$$
$$-1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ 1$$
$$1\ 1\ -1\ 1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1$$
$$1\ -1\ -1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ -1$$
$$-1\ 1\ -1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ -1\ 1\ -1\ -1$$
$$-1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1$$
$$1\ 1\ 1\ -1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ -1\ -1$$
$$1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ -1$$
$$-1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ 1\ 1\ -1\ -1\ 1\ 1$$
$$-1\ -1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ 1\ 1$$
$$1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ -1\ -1\ 1$$
$$-1\ -1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1\ -1\ -1\ -1$$
$$1\ 1\ 1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ -1\ -1\ -1\ 1$$
$$1\ 1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ -1$$
$$1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ -1\ -1$$
$$-1\ -1\ -1\ 1\ -1\ -1\ -1\ -1\ -1\ 1\ 1\ 1\ 1\ -1$$
$$-1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$$
$$-1\ -1\ 1\ -1\ -1\ -1\ -1\ -1\ 1\ 1\ -1\ 1\ -1\ 1$$
$$1\ -1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ 1\ -1\ -1\ -1\ -1$$
$$1\ 1\ 1\ -1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ 1$$
$$1\ -1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ -1\ 1\ -1$$
$$-1\ -1\ 1\ -1\ -1\ 1\ -1\ 1\ 1\ -1\ -1\ -1\ 1\ 1$$
$$-1\ -1\ -1\ -1\ 1\ -1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1$$
$$1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ -1\ -1$$
$$-1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ -1$$
$$1\ -1\ 1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ -1\ -1\ 1\ 1$$
$$1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ 1\ -1\ -1\ -1\ 1\ -1$$
$$1\ -1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ 1\ -1$$
$$1\ -1\ 1\ 1\ 1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ 1$$
$$1\ -1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ -1\ -1\ 1\ -1$$

1 1 -1 -1 -1 1 -1 -1 };

4xEHT_LTF_partB =

{-1 -1 1 -1 1 1 1 1 1 1 -1 -1 -1 -1

1 -1 -1 1 -1 -1 -1 1 1 -1 -1 -1 1 -1

-1 1 -1 1 1 1 -1 1 -1 1 -1 -1 -1 -1

1 1 1 1 1 -1 -1 1 -1 1 -1 -1 -1 1

1 -1 -1 1 -1 -1 -1 1 1 1 -1 -1 1 1

-1 -1 1 -1 1 1 -1 1 -1 1 1 1 -1 1

-1 1 1 1 1 1 1 -1 -1 -1 1 -1 1 -1

-1 -1 1 -1 -1 1 1 1 1 1 1 -1 1 -1

1 1 -1 1 -1 -1 -1 -1 1 1 -1 -1 -1 1

1 -1 1 -1 -1 1 -1 -1 -1 1 -1 1 -1 1

-1 -1 -1 1 -1 1 -1 1 1 1 -1 -1 -1 1

-1 -1 1 1 -1 1 1 1 -1 -1 -1 -1 1 -1

-1 -1 -1 -1 -1 1 -1 1 1 -1 1 -1 1 1

-1 -1 -1 -1 -1 1 -1 -1 1 1 -1 -1 -1 1

-1 -1 1 -1 -1 1 1 -1 -1 1 1 1 1 -1

-1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 -1 1 -1

1 1 -1 1 1 1 1 1 -1 1 1 1 -1 -1

1 -1 1 -1 1 1 1 1 1 -1 -1 -1 1 1

-1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 -1

1 -1 1 -1 -1 1 1 1 -1 1 1 1 1 1

-1 1 1 -1 1 -1 1 -1 -1 -1 -1 -1 1 -1

-1 -1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1

1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1 -1 -1

-1 -1 -1 1 1 -1 1 -1 1 1 -1 1 -1 -1

-1 -1 -1 -1 1 -1 -1 -1 -1 1 1 1 -1 1

1 -1 -1 -1 -1 1 1 -1 1 -1 1 -1 1 -1

1 -1 -1 1 -1 1 -1 -1 -1 -1 -1 -1 1 1

-1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1

-1 -1 1 -1 1 -1 -1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 -1 -1 1 1 1 -1

1 1 -1 -1 1 1 1 -1 1 -1 1 1 -1 -1

-1 -1 -1 1 1 1 1 -1 1 -1 1 -1 -1 -1

1 -1 1 1 -1 1 1 1 -1 -1 1 1 1 -1

1 1 -1 1 1 1 1 -1}.

10. An apparatus for transmitting a physical layer protocol data unit, comprising:

a receiving unit, configured to receive a physical layer protocol data unit, PPDU; and

a parsing unit, configured to parse the received PPDU to obtain an LTF sequence carried in a long training field of the PPDU; and

an estimating unit, configured to estimate a channel according to the obtained LTF sequence and a specified LTF sequence, wherein a specified 320 MHz 4x LTF sequence is $4xEHT_LTF_320M_{-2036:2036}$, $4xEHT_LTF_320M_{-2036:2036}$ = {4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB }, wherein 4xEHT_LTF_partA is a sequence segment including ±1, 4xEHT_LTF_partB is a sequence segment including ±1, 0_5 represents 5 consecutive 0s, 0_{23} represents 23 consecutive 0s.

11. The apparatus according to claim 10, 4xEHT_LTF_partA includes 498 elements, 4xEHT_LTF_partB includes 498 elements.

12. The apparatus according to claim 10 or 11, wherein

4xEHT_LTF_partA =

{ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1

1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1

1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

1 1 1 −1 1 −1 1 1 1 1 1 1 −1 −1

−1 1 −1 1 −1 −1 −1 1 −1 −1 1 1 1 1

1 1 −1 1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 1 1 1 −1 −1 −1 −1 −1 −1 −1

−1 1 −1 −1 1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1

1 1 1 −1 1 1 1 1 1 1 −1 1 −1 −1

1 −1 1 −1 −1 1 1 1 1 1 −1 1 1 −1

−1 1 1 1 −1 1 1 −1 1 1 −1 −1 1 1

−1 −1 −1 −1 1 1 1 1 1 −1 1 1 1 1

1 −1 1 −1 1 −1 −1 1 −1 −1 −1 −1 −1 1

−1 −1 −1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1

1 1 1 1 −1 1 −1 −1 1 1 −1 −1 −1 1

1 1 1 1 −1 1 −1 1 −1 −1 1 1 1 −1

1 1 1 1 1 −1 1 1 −1 1 −1 1 −1 −1

−1 −1 −1 1 −1 −1 −1 −1 −1 1 1 1 1 −1

```
    -1     1     1    -1    -1     1    -1    -1     1    -1    -1    -1     1     1
    -1    -1     1    -1    -1    -1    -1    -1     1     1    -1     1    -1     1
     1    -1     1    -1    -1    -1    -1    -1    -1     1    -1    -1    -1    -1
     1     1     1    -1     1     1    -1    -1     1    -1    -1    -1     1     1
     1    -1     1    -1     1    -1    -1    -1     1    -1     1    -1     1    -1
    -1    -1     1    -1    -1     1    -1     1     1    -1    -1    -1     1     1
    -1    -1    -1    -1     1    -1     1     1    -1     1    -1     1     1     1
     1     1     1    -1    -1     1    -1    -1    -1     1    -1     1    -1    -1
    -1     1     1     1     1     1     1    -1     1    -1     1     1     1    -1
     1    -1     1     1    -1     1    -1    -1     1     1    -1    -1     1     1
     1    -1    -1    -1     1    -1    -1     1     1    -1    -1    -1     1    -1
     1    -1    -1     1     1     1     1     1    -1    -1    -1    -1     1    -1
     1    -1     1     1     1    -1     1    -1    -1     1    -1    -1    -1     1
     1    -1    -1    -1     1    -1    -1     1    -1    -1    -1    -1     1    -1
     1     1    -1    -1    -1     1    -1    -1};

4xEHT_LTF_partB =

{-1    -1     1    -1     1     1     1     1     1     1    -1    -1    -1    -1
     1    -1    -1     1    -1    -1    -1     1     1    -1    -1    -1     1    -1
    -1     1    -1     1     1     1    -1     1    -1     1    -1    -1    -1    -1
     1     1     1     1     1    -1    -1     1    -1     1    -1    -1    -1     1
     1    -1    -1     1    -1    -1    -1     1     1     1    -1    -1     1     1
    -1    -1     1    -1     1     1    -1     1    -1     1     1     1    -1     1
    -1     1     1     1     1     1     1    -1    -1    -1     1    -1     1    -1
    -1    -1     1    -1    -1     1     1     1     1     1     1    -1     1    -1
     1     1    -1     1    -1    -1    -1    -1     1     1    -1    -1    -1     1
     1    -1     1    -1    -1     1    -1    -1    -1     1    -1     1    -1     1
    -1    -1    -1     1    -1     1    -1     1     1     1    -1    -1    -1     1
    -1    -1     1     1    -1     1     1     1    -1    -1    -1    -1     1    -1
    -1    -1    -1    -1    -1     1    -1     1     1    -1     1    -1     1     1
    -1    -1     1    -1    -1     1     1    -1    -1     1     1     1     1    -1
    -1    -1    -1    -1     1    -1    -1    -1    -1    -1     1    -1     1    -1
     1     1    -1     1     1     1     1     1    -1     1     1     1    -1    -1
     1    -1     1    -1     1     1     1     1     1    -1    -1    -1     1     1
    -1    -1    -1    -1    -1    -1    -1    -1     1     1     1     1     1    -1
     1    -1     1    -1    -1     1     1     1    -1     1     1     1     1     1
    -1     1     1    -1     1    -1     1    -1    -1    -1    -1    -1     1    -1
    -1    -1    -1    -1     1     1     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1    -1    -1    -1     1     1    -1    -1     1    -1    -1
    -1    -1    -1     1     1    -1     1    -1     1     1    -1     1    -1    -1
```

−1 −1 −1 −1 1 −1 −1 −1 −1 1 1 1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1

1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1

−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1

−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1

−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1

1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1

−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1

1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1

1 1 −1 1 1 1 1 −1 }；

13. A computer program, wherein the computer program comprises instructions used to perform the method according to any one of claims 1 to 6.

14. A computer-readable storage medium, configured to store a computer program, wherein the computer program comprises instructions used to perform the method according to any one of claims 1 to 6.

15. A method for transmitting a physical layer protocol data unit, comprising:

generating, a physical layer protocol data unit, PPDU, wherein the PPDU comprises a long training field carrying an LTF sequence, a 160 MHz 4x LTF sequence is $4xEHT_LTF_160M_{-1012:1012}$, $4xEHT_LTF_160M_{-1012:1012}$={4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB }, wherein

4xEHT_LTF_partA =

{ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1

1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1

1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

1 1 1 −1 1 −1 1 1 1 1 1 1 −1 −1

−1 1 −1 1 −1 −1 −1 1 −1 −1 1 1 1 1

1 1 −1 1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 1 1 1 −1 −1 −1 −1 −1 −1 −1

−1 1 −1 −1 1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1

1 1 1 −1 1 1 1 1 1 1 −1 1 −1 −1

1 −1 1 −1 −1 1 1 1 1 1 −1 1 1 −1

−1 1 1 1 −1 1 1 −1 1 1 −1 −1 1 1

−1 −1 −1 −1 1 1 1 1 1 −1 1 1 1 1

1 −1 1 −1 1 −1 −1 1 −1 −1 −1 −1 −1 1

−1 −1 −1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1

1 1 1 1 −1 1 −1 −1 1 1 −1 −1 −1 1

1 1 1 1 −1 1 −1 1 −1 −1 1 1 1 −1

1 1 1 1 1 −1 1 1 −1 1 −1 1 −1 −1

−1 −1 −1 1 −1 −1 −1 −1 −1 1 1 1 1 −1

−1 1 1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 1 −1 −1 −1 −1 −1 1 1 −1 1 −1 1

1 −1 1 −1 −1 −1 −1 −1 −1 1 −1 −1 −1 −1

1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 −1 1 1 −1 −1 −1 1 1

−1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1 1

1 1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 −1

−1 1 1 1 1 1 1 −1 1 −1 1 1 1 −1

1 −1 1 1 −1 1 −1 −1 1 1 −1 −1 1 1

1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1 −1

1 −1 −1 1 1 1 1 1 −1 −1 −1 −1 1 −1

1 −1 1 1 1 −1 1 −1 −1 1 −1 −1 −1 1

1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 −1 1 −1

1 1 −1 −1 −1 1 −1 −1 };

4xEHT_LTF_partB =

{−1 −1 1 −1 1 1 1 1 1 1 −1 −1 −1 −1

1 −1 −1 1 −1 −1 −1 1 1 −1 −1 −1 1 −1

−1 1 −1 1 1 1 −1 1 −1 1 −1 −1 −1 −1

1 1 1 1 1 −1 −1 1 −1 1 −1 −1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 −1 1 1

−1 −1 1 −1 1 1 −1 1 −1 1 1 1 −1 1

−1 1 1 1 1 1 1 −1 −1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 1 1 1 1 1 −1 1 −1

1 1 −1 1 −1 −1 −1 −1 1 1 −1 −1 −1 1

1 −1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 1

−1 −1 −1 1 −1 1 −1 1 1 1 −1 −1 −1 1

−1 −1 1 1 −1 1 1 1 −1 −1 −1 −1 1 −1

−1 −1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1

−1 −1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1

−1 −1 1 −1 −1 1 1 −1 −1 1 1 1 1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 −1 1 −1 1 −1

1 1 −1 1 1 1 1 1 −1 1 1 1 −1 −1
1 −1 1 −1 1 1 1 1 1 −1 −1 −1 1 1
−1 −1 −1 −1 −1 −1 −1 −1 1 1 1 1 1 −1
1 −1 1 −1 −1 1 1 1 −1 1 1 1 1 1
−1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1 1 −1
−1 −1 −1 −1 1 1 1 1 −1 −1 1 1 −1 −1
1 −1 −1 1 −1 −1 −1 1 1 −1 −1 1 −1 −1
−1 −1 −1 1 1 −1 1 −1 1 1 −1 1 −1 −1
−1 −1 −1 −1 1 −1 −1 −1 −1 1 1 1 −1 1
1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1
−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1
1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1
1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1
−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1
−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1
−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1
1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1
−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1
1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1
1 1 −1 1 1 1 1 −1 }；

0_5 represents 5 consecutive 0s；

0_{23} represents 23 consecutive 0s;

sending the PPDU.

16. A method for transmitting a physical layer protocol data unit, comprising:

receiving a physical layer protocol data unit, PPDU; and

parsing the received PPDU to obtain an LTF sequence carried in a long training field of the PPDU; and

estimating a channel according to the obtained LTF sequence and a specified LTF sequence, wherein a specified 160 MHz 4x LTF sequence is $4xEHT_LTF_160M_{-1012:1012}$, $4xEHT_LTF_160M_{-1012:1012}$={4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB }, wherein

4xEHT_LTF_partA =

{ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1
−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1
1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1
1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1
1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

```
    1   1   1  -1   1  -1   1   1   1   1   1   1  -1  -1
   -1   1  -1   1  -1  -1  -1   1  -1  -1   1   1   1   1
    1   1  -1   1  -1   1   1  -1   1  -1   1  -1   1  -1
    1  -1  -1   1   1   1   1  -1  -1  -1  -1  -1  -1  -1
   -1   1  -1  -1   1  -1  -1   1   1   1  -1   1  -1  -1
   -1   1   1   1  -1   1   1  -1  -1   1  -1  -1  -1   1
    1   1   1  -1   1   1   1   1   1   1  -1   1  -1  -1
    1  -1   1  -1  -1   1   1   1   1   1  -1   1   1  -1
   -1   1   1   1  -1   1   1  -1   1   1  -1  -1   1   1
   -1  -1  -1  -1   1   1   1   1   1  -1   1   1   1   1
    1  -1   1  -1   1  -1  -1   1  -1  -1  -1  -1  -1   1
   -1  -1  -1   1   1  -1   1  -1   1  -1  -1  -1  -1  -1
    1   1   1   1  -1   1  -1  -1   1   1  -1  -1  -1   1
    1   1   1   1  -1   1  -1   1  -1  -1   1   1   1  -1
    1   1   1  -1   1   1  -1  -1   1  -1  -1  -1   1   1
   -1  -1  -1   1  -1  -1  -1  -1  -1   1   1   1   1  -1
   -1   1   1  -1  -1   1  -1  -1   1  -1  -1  -1   1   1
    1  -1   1  -1  -1  -1  -1  -1  -1   1  -1  -1  -1  -1
    1   1  -1  -1   1  -1   1   1  -1  -1   1  -1   1  -1
    1  -1   1  -1   1  -1  -1  -1   1  -1   1  -1   1  -1
   -1  -1   1  -1  -1   1  -1   1   1  -1  -1  -1   1   1
   -1  -1  -1  -1   1  -1   1   1  -1   1  -1   1   1   1
    1   1   1  -1  -1   1  -1  -1  -1   1  -1   1  -1  -1
   -1   1   1   1   1   1   1  -1   1  -1   1   1   1  -1
    1  -1   1   1  -1   1  -1  -1   1   1  -1  -1   1   1
    1  -1  -1  -1   1  -1  -1   1   1  -1  -1  -1   1  -1
    1  -1  -1   1   1   1   1   1  -1  -1  -1  -1   1  -1
    1  -1   1   1   1  -1   1  -1  -1   1  -1  -1  -1   1
    1  -1  -1  -1   1  -1  -1   1  -1  -1  -1  -1   1  -1
    1   1  -1  -1  -1   1  -1  -1};
```

```
4xEHT_LTF_partB =
{-1  -1   1  -1   1   1   1   1   1   1  -1  -1  -1  -1
   1  -1  -1   1  -1  -1  -1   1   1  -1  -1  -1   1  -1
  -1   1  -1   1   1   1  -1   1  -1   1  -1  -1  -1  -1
   1   1   1   1   1  -1  -1   1  -1   1  -1  -1  -1   1
   1  -1  -1   1  -1  -1  -1   1   1   1  -1  -1   1   1
  -1  -1   1  -1   1   1  -1   1  -1   1   1   1  -1   1
  -1   1   1   1   1   1   1  -1  -1  -1   1  -1   1  -1
  -1  -1   1  -1  -1   1   1   1   1   1   1  -1   1  -1
```

1 1 −1 1 −1 −1 −1 −1 1 1 −1 −1 −1 1

1 −1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 1

−1 −1 −1 1 −1 1 −1 1 1 1 −1 −1 −1 1

−1 −1 1 1 −1 1 1 1 −1 −1 −1 −1 1 −1

−1 −1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1

−1 −1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1

−1 −1 1 −1 −1 1 1 −1 −1 1 1 1 1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 −1 1 −1 1 −1

1 1 −1 1 1 1 1 1 −1 1 1 1 −1 −1

1 −1 1 −1 1 1 1 1 1 −1 −1 −1 1 1

−1 −1 −1 −1 −1 −1 −1 −1 1 1 1 1 1 −1

1 −1 1 −1 −1 1 1 1 −1 1 1 1 1 1

−1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1 1 −1

−1 −1 −1 −1 1 1 1 1 −1 −1 1 1 −1 −1

1 −1 −1 1 −1 −1 −1 1 1 −1 −1 1 −1 −1

−1 −1 −1 1 1 −1 1 −1 1 1 −1 1 −1 −1

−1 −1 −1 −1 1 −1 −1 −1 −1 1 1 1 −1 1

1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1

1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1

−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1

−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1

−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1

1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1

−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1

1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1

1 1 −1 1 1 1 1 −1 }；

0_5 represents 5 consecutive 0s；

0_{23} represents 23 consecutive 0s.

17. An apparatus, configured to perform a method as claimed in any one of claims 15 to 16.

18. A program which makes a computer to perform a method as claimed in any one of claims 15 to 16.

19. A computer-readable recording medium on which a program is recorded, wherein the program, when executed, enables a computer to perform a method as claimed in any one of claims 15 to 16.

20. A method for transmitting a physical layer protocol data unit, comprising:

generating, a physical layer protocol data unit, PPDU, wherein the PPDU comprises a long training field carrying an LTF sequence, a 80 MHz 4x LTF sequence is $4xEHT_LTF_80M_{-500:500}$, $4xEHT_LTF_80M_{-500:500} = \{ 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB \}$, wherein

$4xEHT_LTF_partA =$

$\{$ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1

1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1

1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

1 1 1 −1 1 −1 1 1 1 1 1 1 −1 −1

−1 1 −1 1 −1 −1 −1 1 −1 −1 1 1 1 1

1 1 −1 1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 1 1 1 −1 −1 −1 −1 −1 −1 −1

−1 1 −1 −1 1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1

1 1 1 −1 1 1 1 1 1 1 −1 1 −1 −1

1 −1 1 −1 −1 1 1 1 1 1 −1 1 1 −1

−1 1 1 1 −1 1 1 −1 1 1 −1 −1 1 1

−1 −1 −1 −1 1 1 1 1 1 −1 1 1 1 1

1 −1 1 −1 1 −1 −1 1 −1 −1 −1 −1 −1 1

−1 −1 −1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1

1 1 1 1 −1 1 −1 −1 1 1 −1 −1 −1 1

1 1 1 1 −1 1 −1 1 −1 −1 1 1 1 −1

1 1 1 1 1 −1 1 1 −1 1 −1 1 −1 −1

−1 −1 −1 1 −1 −1 −1 −1 −1 1 1 1 1 −1

−1 1 1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 1 −1 −1 −1 −1 −1 1 1 −1 1 −1 1

1 −1 1 −1 −1 −1 −1 −1 −1 1 −1 −1 −1 −1

1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 −1 1 1 −1 −1 −1 1 1

−1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1 1

1 1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 −1

−1 1 1 1 1 1 1 −1 1 −1 1 1 1 −1

1 −1 1 1 −1 1 −1 −1 1 1 −1 −1 1 1

1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1 −1

1 −1 −1 1 1 1 1 1 −1 −1 −1 −1 1 −1

1 −1 1 1 1 −1 1 −1 −1 1 −1 −1 −1 1

1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 −1 1 −1

1 1 −1 −1 −1 1 −1 −1 };

4xEHT_LTF_partB =

{−1 −1 1 −1 1 1 1 1 1 1 −1 −1 −1 −1
1 −1 −1 1 −1 −1 −1 1 1 −1 −1 −1 1 −1
−1 1 −1 1 1 1 −1 1 −1 1 −1 −1 −1 −1
1 1 1 1 1 −1 −1 1 −1 1 −1 −1 −1 1
1 −1 −1 1 −1 −1 −1 1 1 1 −1 −1 1 1
−1 −1 1 −1 1 1 −1 1 −1 1 1 1 −1 1
−1 1 1 1 1 1 1 −1 −1 −1 1 −1 1 −1
−1 −1 1 −1 −1 1 1 1 1 1 1 −1 1 −1
1 1 −1 1 −1 −1 −1 −1 1 1 −1 −1 −1 1
1 −1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 1
−1 −1 −1 1 −1 1 −1 1 1 1 −1 −1 −1 1
−1 −1 1 1 −1 1 1 1 −1 −1 −1 −1 1 −1
−1 −1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1
−1 −1 −1 −1 −1 1 −1 −1 1 1 −1 −1 −1 1
−1 −1 1 −1 −1 1 1 −1 −1 1 1 1 1 −1
−1 −1 −1 −1 1 −1 −1 −1 −1 −1 1 −1 1 −1
1 1 −1 1 1 1 1 1 −1 1 1 1 −1 −1
1 −1 1 −1 1 1 1 1 1 −1 −1 −1 1 1
−1 −1 −1 −1 −1 −1 −1 −1 1 1 1 1 1 −1
1 −1 1 −1 −1 1 1 1 −1 1 1 1 1 1
−1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1 1 −1
−1 −1 −1 −1 1 1 1 1 −1 −1 1 1 −1 −1
1 −1 −1 1 −1 −1 −1 1 1 1 −1 1 −1 −1
−1 1 1 −1 1 1 −1 1 1 1 1 1 1 1
1 −1 −1 −1 −1 1 1 −1 1 −1 1 −1 1 −1
1 −1 −1 1 −1 1 −1 −1 −1 −1 −1 −1 1 1
−1 1 1 1 −1 1 −1 1 1 1 −1 −1 −1 −1
−1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 −1 1
−1 1 1 −1 −1 1 1 −1 −1 −1 1 1 1 −1
1 1 −1 −1 1 1 1 −1 1 −1 1 1 −1 −1
−1 −1 −1 1 1 1 1 −1 1 −1 1 −1 −1 −1
1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1
1 1 −1 1 1 1 1 −1 };

0_5 represents 5 consecutive 0s;

sending the PPDU.

21. A method for transmitting a physical layer protocol data unit, comprising:

receiving a physical layer protocol data unit, PPDU; and

parsing the received PPDU to obtain an LTF sequence carried in a long training field of the PPDU; and

estimating a channel according to the obtained LTF sequence and a specified LTF sequence, wherein a specified 80 MHz 4x LTF sequence is $4xEHT_LTF_80M_{-500:500}$, $4xEHT_LTF_80M_{-500:500} = \{ 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB \}$, wherein

$4xEHT_LTF_partA =$

$\{$ 1 −1 −1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 −1 1 −1 −1 1 −1 1 1 1 −1 1 −1

1 −1 −1 −1 −1 1 1 1 1 1 −1 −1 1 −1

1 −1 −1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 −1 1 1 −1 −1 1 −1 1 1 −1 1 −1

1 1 1 −1 1 −1 1 1 1 1 1 1 −1 −1

−1 1 −1 1 −1 −1 −1 1 −1 −1 1 1 1 1

1 1 −1 1 −1 1 1 −1 1 −1 1 −1 1 −1

1 −1 −1 1 1 1 1 −1 −1 −1 −1 −1 −1 −1

−1 1 −1 −1 1 −1 −1 1 1 1 −1 1 −1 −1

−1 1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1

1 1 1 −1 1 1 1 1 1 1 −1 1 −1 −1

1 −1 1 −1 −1 1 1 1 1 1 −1 1 1 −1

−1 1 1 1 −1 1 1 −1 1 1 −1 −1 1 1

−1 −1 −1 −1 1 1 1 1 1 −1 1 1 1 1

1 −1 1 −1 1 −1 −1 1 −1 −1 −1 −1 −1 1

−1 −1 −1 1 1 −1 1 −1 1 −1 −1 −1 −1 −1

1 1 1 1 −1 1 −1 −1 1 1 −1 −1 −1 1

1 1 1 1 −1 1 −1 1 −1 −1 1 1 1 −1

1 1 1 1 1 −1 1 1 −1 1 −1 1 −1 −1

−1 −1 −1 1 −1 −1 −1 −1 −1 1 1 1 1 −1

−1 1 1 −1 −1 1 −1 −1 1 −1 −1 −1 1 1

−1 −1 1 −1 −1 −1 −1 −1 1 1 −1 1 −1 1

1 −1 1 −1 −1 −1 −1 −1 −1 1 −1 −1 −1 −1

1 1 1 −1 1 1 −1 −1 1 −1 −1 −1 1 1

1 −1 1 −1 1 −1 −1 −1 1 −1 1 −1 1 −1

−1 −1 1 −1 −1 1 −1 1 1 −1 −1 −1 1 1

−1 −1 −1 −1 1 −1 1 1 −1 1 −1 1 1 1

1 1 1 −1 −1 1 −1 −1 −1 1 −1 1 −1 −1

−1 1 1 1 1 1 1 −1 1 −1 1 1 1 −1

1 −1 1 1 −1 1 −1 −1 1 1 −1 −1 1 1

```
    1   -1   -1   -1    1   -1   -1    1    1   -1   -1   -1    1   -1
    1   -1   -1    1    1    1    1    1   -1   -1   -1   -1    1   -1
    1   -1    1    1    1   -1    1   -1   -1    1   -1   -1   -1    1
    1   -1   -1   -1    1   -1   -1    1   -1   -1   -1   -1    1   -1
    1    1   -1   -1   -1    1   -1   -1};

4xEHT_LTF_partB =

{-1   -1    1   -1    1    1    1    1    1    1   -1   -1   -1   -1
    1   -1   -1    1   -1   -1   -1    1    1   -1   -1   -1    1   -1
   -1    1   -1    1    1    1   -1    1   -1    1   -1   -1   -1   -1
    1    1    1    1    1   -1   -1    1   -1    1   -1   -1   -1    1
    1   -1   -1    1   -1   -1   -1    1    1    1   -1   -1    1    1
   -1   -1    1   -1    1    1   -1    1   -1    1    1    1   -1    1
   -1    1    1    1    1    1    1   -1   -1   -1    1   -1    1   -1
   -1   -1    1   -1   -1    1    1    1    1    1    1   -1    1   -1
    1    1   -1    1   -1   -1   -1   -1    1    1   -1   -1   -1    1
    1   -1    1   -1   -1    1   -1   -1   -1    1   -1    1   -1    1
   -1   -1   -1    1   -1    1   -1    1    1    1   -1   -1   -1    1
   -1   -1    1    1   -1    1    1    1   -1   -1   -1   -1    1   -1
   -1   -1   -1   -1   -1    1   -1    1    1   -1    1   -1    1    1
   -1   -1   -1   -1   -1    1   -1   -1    1    1   -1   -1   -1    1
   -1   -1    1   -1   -1    1    1   -1   -1    1    1    1    1   -1
   -1   -1   -1   -1    1   -1   -1   -1   -1   -1    1   -1    1   -1
    1    1   -1    1    1    1    1    1   -1    1    1    1   -1   -1
    1   -1    1   -1    1    1    1    1    1   -1   -1   -1    1    1
   -1   -1   -1   -1   -1   -1   -1   -1    1    1    1    1    1   -1
    1   -1    1   -1   -1    1    1    1   -1    1    1    1    1    1
   -1    1    1   -1    1   -1    1   -1   -1   -1   -1   -1    1   -1
   -1   -1   -1   -1    1    1    1    1   -1   -1    1    1   -1   -1
    1   -1   -1    1   -1   -1   -1    1    1   -1   -1    1   -1   -1
    1   -1   -1    1    1    1    1   -1    1   -1    1   -1   -1   -1
   -1   -1   -1    1    1    1    1   -1    1   -1    1   -1   -1   -1
```

1 −1 1 1 −1 1 1 1 −1 −1 1 1 1 −1

1 1 −1 1 1 1 1 −1 };

0_5 represents 5 consecutive 0s.

22. An apparatus, configured to perform a method as claimed in any one of claims 20 to 21.

23. A program which makes a computer to perform a method as claimed in any one of claims 20 to 21.

24. A computer-readable recording medium on which a program is recorded, wherein the program, when executed, enables a computer to perform a method as claimed in any one of claims 20 to 21.

~~3. The method according to claim 1 or 2, wherein~~

~~an 80 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_80M$_{-500:500}$ = {4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}; or~~

~~a 160 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_160M$_{-1012:1012}$ =~~

~~{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}; or~~

~~a 320 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_320M$_{-2036:2036}$ =~~

~~{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB};~~

~~herein, 0_{23} represents 23 consecutive 0s, and 0_5 represents 5 consecutive 0s;~~

~~4xEHT_LTF_partA =~~

~~{1, −1, −1, −1, −1, 1, −1, −1, 1, −1, −1, −1, 1, 1~~

~~−1, 1, −1, 1, −1, 1, −1, 1, 1, 1, −1, 1, −1~~

~~−1, −1, −1, −1, −1, 1, 1, 1, 1, 1, −1, −1, 1, −1~~

~~−1, 1, −1, 1, 1, 1, −1, 1, 1, −1, 1, −1, 1, 1~~

~~−1, −1, −1, 1, 1, −1, −1, 1, −1, 1, 1, −1, 1, −1~~

~~1, 1, 1, −1, 1, −1, 1, 1, 1, 1, 1, 1, −1, −1~~

~~−1, 1, −1, 1, −1, −1, 1, 1, −1, 1, 1, 1, 1, 1~~

~~−1, 1, −1, 1, −1, 1, 1, −1, 1, −1, 1, −1, 1, −1~~

~~−1, 1, −1, 1, 1, 1, 1, 1, −1, 1, −1, 1, −1, −1~~

~~-1, 1, -1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1~~

~~-1, 1, 1, 1, -1, 1, 1, -1, -1, 1, -1, -1, -1, 1~~

~~-1, 1, 1, -1, 1, 1, 1, 1, 1, 1, -1, 1, -1, -1~~

~~-1, -1, 1, -1, -1, 1, 1, 1, 1, 1, -1, 1, 1, -1~~

~~-1, 1, 1, 1, -1, 1, 1, -1, 1, 1, -1, -1, 1, 1~~

~~-1, -1, -1, 1, 1, 1, 1, 1, 1, -1, 1, 1, 1, 1~~

~~-1, -1, 1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, 1~~

~~-1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1~~

~~-1, 1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1~~

~~-1, 1, 1, 1, 1, -1, 1, 1, -1, 1, -1, 1, 1, -1~~

~~-1, -1, -1, -1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1~~

~~-1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, 1~~

~~-1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, -1, -1, -1, 1, 1, 1, -1, 1, -1, 1, 1, 1, 1~~

~~-1, 1, -1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, 1~~

~~-1, 1, 1, 1, 1, -1, 1, 1, -1, 1, -1, 1, 1, -1~~

~~-1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, 1, 1~~

~~-1, 1, -1, 1, -1, 1, 1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, -1, 1, 1, 1, 1, -1, 1, -1, 1, 1, 1, -1, -1~~

~~-1, 1, -1, -1, 1, -1, 1, 1, 1, 1, 1, -1, -1, -1~~

~~-1, 1, 1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, 1~~

~~-1, 1, -1, 1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1~~

~~-1, 1, -1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, 1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1~~

~~-1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, 1, 1~~

~~-1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1~~

~~-1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, 1~~

~~-1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, -1, -1, -1, 1, 1, 1, -1, 1, -1, 1, 1, 1, 1~~

~~-1, 1, -1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, 1~~

~~-1, 1, 1, 1, 1, -1, 1, 1, -1, 1, -

~~−1, 1, 1, 1, 1, −1, −1, 1, −1, 1, −1, −1, −1, 1~~

~~−1, −1, −1, 1, −1, −1, −1, 1, 1, 1, −1, −1, 1, 1~~

~~−1, −1, 1, −1, 1, 1, −1, 1, −1, 1, 1, 1, −1, 1~~

~~−1, 1, 1, 1, 1, 1, 1, −1, −1, −1, 1, −1, 1, −1~~

5 ~~−1, −1, 1, −1, −1, 1, 1, 1, 1, 1, 1, −1, 1, −1~~

~~−1, 1, −1, 1, −1, −1, −1, 1, 1, 1, −1, −1, −1, 1~~

~~−1, −1, 1, −1, −1, 1, −1, −1, −1, 1, −1, 1, −1, 1~~

~~−1, −1, −1, 1, −1, 1, −1, 1, 1, 1, −1, −1, −1, 1~~

~~−1, −1, 1, 1, −1, 1, 1, 1, −1, −1, −1, −1, 1, −1~~

10 ~~−1, −1, −1, −1, −1, 1, −1, 1, 1, −1, 1, −1, 1, 1~~

~~−1, −1, −1, −1, 1, 1, −1, −1, 1, 1, −1, −1, −1, 1~~

~~−1, −1, 1, −1, −1, 1, 1, −1, −1, 1, 1, 1, 1, −1~~

~~−1, −1, −1, −1, 1, −1, −1, −1, −1, 1, 1, −1, 1, −1~~

~~−1, 1, −1, 1, 1, 1, 1, 1, −1, 1, 1, 1, −1, −1~~

15 ~~−1, −1, 1, −1, 1, 1, 1, 1, 1, −1, −1, −1, 1, 1~~

~~−1, −1, −1, −1, −1, −1, −1, −1, 1, 1, 1, 1, 1, −1~~

~~−1, −1, 1, −1, −1, 1, 1, 1, −1, 1, 1, 1, 1, 1~~

~~−1, 1, 1, −1, 1, −1, 1, −1, −1, −1, −1, −1, 1, −1~~

~~−1, −1, −1, −1, 1, 1, 1, 1, −1, −1, 1, 1, −1, −1~~

20 ~~−1, −1, −1, 1, −1, −1, 1, 1, 1, −1, −1, 1, −1, −1~~

~~−1, −1, −1, 1, 1, −1, 1, −1, 1, 1, −1, 1, −1, −1~~

~~−1, −1, −1, −1, 1, −1, −1, −1, −1, 1, 1, 1, −1, 1~~

~~−1, −1, −1, 1, −1, −1, 1, 1, 1, 1, −1, 1, −1, −1~~

~~−1, 1, 1, −1, 1, 1, −1, 1, 1, 1, 1, 1, 1, 1~~

25 ~~−1, −1, −1, −1, −1, 1, 1, −1, 1, −1, 1, −1, 1, −1~~

~~−1, −1, −1, 1, −1, 1, −1, −1, −1, −1, −1, −1, 1, 1~~

~~−1, 1, 1, 1, −1, 1, −1, 1, 1, 1, −1, −1, −1, −1~~

~~−1, −1, 1, −1, 1, 1, −1, −1, 1, −1, 1, −1, 1, 1~~

~~−1, 1, 1, −1, −1, 1, 1, −1, −1, −1, 1, 1, 1, −1~~

30 ~~−1, 1, −1, −1, 1, 1, 1, −1, 1, −1, 1, 1, −1, −1~~

~~−1, −1, −1, 1, 1, 1, 1, −1, 1, −1, 1, −1, −1, −1~~

~~−1, −1, 1, 1, −1, 1, 1, 1, −1, −1, 1, 1, 1, −1~~

~~−1, 1, −1, 1, 1, 1, 1, −1}.~~

~~4. The method according to claim 1 or 2, wherein~~

~~an 80 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_80M$_{-500:500}$ =~~

~~{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB}; or~~

~~a 160 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_160M$_{-1012:1012}$ =~~

~~{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, −4xEHT_LTF_partB}; or~~

~~a 320 MHz 4x LTF sequence is:~~

~~4xEHT_LTF_320M$_{-2036:2036}$ =~~

~~{4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, 4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB, 0_{23}, −4xEHT_LTF_partA, 0_5, 4xEHT_LTF_partB};~~

~~herein, 0_{23} represents 23 consecutive 0s, and 0_5 represents 5 consecutive 0s;~~

~~4xEHT_LTF_partA =~~

~~{ 1, −1, −1, 1, 1, −1, 1, 1, 1, −1, −1, 1, 1, 1~~

~~−1, 1, −1, 1, 1, −1, −1, −1, −1, −1, 1, −1, 1, −1~~

~~−1, 1, 1, −1, 1, −1, 1, −1, 1, 1, 1, −1, 1, −1, −1~~

~~−1, 1, 1, −1, 1, −1, −1, −1, −1, −1, −1, −1, 1, 1~~

~~−1, −1, −1, 1, 1, −1, −1, −1, −1, −1, 1, 1, 1, −1~~

~~−1, −1, 1, −1, 1, −1, 1, −1, 1, −1, 1, −1, −1, −1, −1~~

~~−1, 1, −1, −1, −1, 1, −1, −1, −1, 1, −1, 1, −1, −1~~

~~−1, 1, 1, 1, 1, 1, 1, 1, 1, −1, 1, 1, 1, −1, 1~~

~~−1, 1, −1, −1, 1, 1, 1, −1, 1, −1, 1, 1, 1, 1, −1~~

~~−1, −1, 1, −1, −1, −1, −1, −1, −1, 1, 1, −1, −1, 1, −1~~

~~−1, 1, −1, 1, 1, 1, 1, 1, −1, −1, 1, −1, −1, −1, −1~~

~~−1, −1, 1, −1, 1, −1, −1, 1, 1, 1, −1, 1, 1, −1, −1~~

~~-1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, 1, -1, 1~~

~~-1, 1, -1, -1, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1~~

~~-1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, -1~~

~~-1, 1, 1, 1, 1, 1, -1, 1, -1, -1, 1, 1, -1, -1~~

~~-1, 1, 1, -1, 1, -1, 1, 1, -1, -1, 1, 1, 1, 1~~

~~-1, 1, -1, -1, 1, -1, 1, -1, -1, 1, -1, 1, 1, 1~~

~~-1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1~~

~~-1, 1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1~~

~~-1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1~~

~~-1, 1, 1, 1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1~~

~~-1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1~~

~~-1, -1, -1, 1, 1, 1, 1, 1, -1, -1, 1, 1, 1, -1~~

~~-1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1, -1, 1~~

~~-1, 1, -1, -1, -1, -1, 1, -1, 1, -1, 1, 1, 1, -1~~

~~-1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1, 1, -1, -1~~

~~-1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, 1, 1, -1~~

~~-1, -1, 1, -1, 1, 1, 1, 1};~~

~~4xEHT_LTF_partB =~~

~~{1, 1, 1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1, -1~~

~~-1, 1, -1, 1, 1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1~~

~~-1, 1, 1, 1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1~~

~~-1, 1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1, 1, -1~~

~~-1, 1, 1, 1, 1, 1, 1, -1, 1, -1, 1, 1, 1, 1~~

~~-1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1~~

~~-1, 1, -1, 1, 1, 1, -1, 1, 1, 1, 1, 1, -1, 1~~

~~-1, 1, -1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1, 1~~

~~−1, −1, −1, 1, 1, −1, −1, −1, −1, −1, −1, −1, 1, 1, −1~~

~~−1, −1, 1, −1, 1, 1, 1, −1, −1, 1, −1, 1, 1, −1~~

~~−1, 1, −1, −1, −1, 1, 1, −1, −1, −1, 1, −1, −1, 1~~

~~−1, −1, 1, −1, 1, 1, 1, −1, −1, 1, −1, 1, −1, −1~~

~~−1, 1, −1, −1, 1, −1, −1, 1, −1, −1, −1, −1, 1, 1~~

~~−1, 1, −1, −1, 1, −1, −1, −1, −1, 1, 1, −1, 1, 1~~

~~−1, −1, 1, 1, −1, 1, 1, −1, −1, 1, −1, 1, −1, −1~~

~~−1, −1, −1, −1, −1, −1, −1, 1, 1, 1, 1, −1, −1, −1~~

~~−1, 1, −1, 1, −1, 1, 1, 1, −1, −1, 1, 1, −1, −1~~

~~−1, 1, −1, −1, −1, 1, −1, −1, −1, −1, −1, 1, 1, −1~~

~~−1, −1, 1, 1, 1, 1, 1, −1, −1, −1, 1, 1, 1, 1~~

~~−1, 1, −1, 1, 1, −1, 1, −1, 1, 1, −1, −1, 1, −1~~

~~−1, 1, −1, 1, −1, −1, −1, 1, 1, 1, −1, 1, −1, 1~~

~~−1, −1, −1, −1, −1, −1, 1, 1, 1, 1, −1, 1, −1, 1~~

~~−1, 1, −1, 1, 1, 1, −1, 1, 1, 1, −1, 1, −1, 1~~

~~−1, −1, 1, 1, −1, 1, 1, 1, 1, 1, −1, −1, 1, 1~~

~~−1, 1, 1, −1, −1, 1, 1, 1, −1, −1, 1, 1, 1, 1~~

~~−1, 1, −1, 1, 1, −1, 1, −1, 1, −1, 1, −1, 1, −1~~

~~−1, 1, 1, 1, −1, −1, 1, 1, −1, 1, −1, 1, −1, 1~~

~~−1, −1, 1, 1, −1, 1, 1, −1, −1, 1, 1, −1, −1, 1~~

~~−1, 1, 1, −1, −1, 1, 1, −1, 1, −1, 1, −1, −1, 1~~

~~−1, 1, −1, 1, 1, −1, −1, −1, 1, 1, 1, −1, 1, −1~~

~~−1, −1, 1, 1, 1, −1, −1, −1, 1, −1, 1, −1, 1, 1~~

~~−1, 1, 1, −1, 1, −1, 1, −1, 1, −1, 1, 1, 1, −1~~

~~−1, −1, 1, −1, −1, 1, 1, 1, −1, −1, 1, 1, 1, 1~~

~~−1, 1, 1, −

~~{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}; or~~

~~a 160 MHz 2x LTF sequence is:~~

~~$2xEHT_LTF_160M_{-1012:1012} =$~~

~~{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, −2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}; or~~

~~a 320 MHz 2x LTF sequence is:~~

~~$2xEHT_LTF_320M_{-2036:2036} =$~~

~~{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, −2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, −2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB};~~

~~herein, 0_5 represents 5 consecutive 0s, and 0_{23} represents 23 consecutive 0s;~~

~~2xEHT_LTF_partA =~~

~~{ 1, 0, −1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0~~

~~−1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, −1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, −1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0, 1, 0~~

~~−1, 0, 1, 0, 1, 0, −1, 0, 1, 0, 1, 0, 1, 0~~

~~−1, 0, 1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0~~

~~−1, 0, −1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0~~

~~−1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1, 0, 1, 0~~

~~−1, 0, −1, 0, 1, 0, 1, 0, −1, 0, −1, 0, −1, 0~~

~~−1, 0, −1, 0, 1, 0, 1, 0, −1, 0, −1, 0, −1, 0~~

~~-1, 0, -1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0, 1, 0~~

~~-1, 0, 1, 0, -1, 0, -1, 0, 1, 0, -1, 0, -1, 0~~

~~-1, 0, -1, 0, -1, 0, 1, 0, 1, 0, -1, 0, 1, 0~~

~~-1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, 1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1, 0~~

~~-1, 0, -1, 0, 1, 0, 1, 0, -1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, 1, 0, 1, 0, 1, 0, 1, 0, -1, 0~~

~~-1, 0, -1, 0, 1, 0, -1, 0, -1, 0, -1, 0, -1, 0~~

~~-1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0~~

~~-1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0~~

~~-1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0~~

~~-1, 0, 1, 0, 1, 0, -1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0~~

~~-1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0~~

~~-1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0~~

~~-1, 0, -1, 0, -1, 0, 1, 0};~~

~~2xEHT_LTF_partB =~~

~~{0, -1, 0, 1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1~~

~~0, -1, 0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0, 1~~

~~0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1~~

~~0, 1, 0, -1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1~~

~~0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, 1~~

~~0, -1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1~~

~~0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, −1, 0, 1~~

~~0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1~~

~~0, 1, 0, 1, 0, −1, 0, −1, 0, 1, 0, 1, 0, 1~~

~~0, −1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1~~

~~0, 1, 0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0, 1, 0, −1~~

~~0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, −1, 0, −1, 0, 1, 0, 1, 0, 1, 0, −1~~

~~0, −1, 0, 1, 0, 1, 0, 1, 0, −1, 0, 1, 0, 1~~

~~0, −1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1~~

~~0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1~~

~~0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, −1, 0, −1, 0, 1, 0, 1, 0, 1, 0, −1~~

~~0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0, −1, 0, 1~~

~~0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0, 1~~

~~0, −1, 0, 1, 0, −1, 0, 1, 0, 1, 0, 1, 0, −1~~

~~0, 1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0, −1, 0, 1~~

~~0, −1, 0, 1, 0, −1, 0, 1, 0, 1, 0, 1, 0, −1~~

~~0, 1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1}.~~

~~6. The method according to claim 1 or 2, wherein~~

~~an 80 MHz 2x LTF sequence is:~~

~~2xEHT_LTF_80M$_{-500:500}$ =~~

~~{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}; or~~

~~a 160 MHz 2x LTF sequence is:~~

~~2xEHT_LTF_160M$_{-1012:1012}$ =~~

~~{2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB, 0_{23}, 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB}; or~~

~~2xEHT_LTF_320M_2036:2036 =~~

~~{2xEHT_LTF_partA, 0_5 , −2xEHT_LTF_partB, 0_{23} , −2xEHT_LTF_partA, 0_5 , −2xEHT_LTF_partB, 0_{23} , 2xEHT_LTF_partA, 0_5 , 2xEHT_LTF_partB, 0_{23} , 2xEHT_LTF_partA, 0_5, 2xEHT_LTF_partB};~~

~~herein, 0_5 represents 5 consecutive 0s, and 0_{23} represents 23 consecutive 0s;~~

~~2xEHT_LTF_partA =~~

~~{ 1, 0, −1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0~~

~~−1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, −1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, −1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0~~

~~−1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0~~

~~−1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, −1, 0, −1, 0, 1, 0, −1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0~~

~~−1, 0, −1, 0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0~~

~~−1, 0, −1, 0, 1, 0, 1, 0, 1, 0, −1, 0, −1, 0~~

~~−1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0~~

~~−1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0~~

~~−1, 0, 1, 0, 1, 0, −1, 0, 1, 0, 1, 0, −1, 0~~

~~-1, 0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0, -1, 0~~

~~-1, 0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0, 1, 0~~

~~-1, 0, 1, 0, -1, 0, 1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0~~

~~-1, 0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0~~

~~-1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0~~

~~-1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0~~

~~-1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0~~

~~-1, 0, 1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0~~

~~-1, 0, 1, 0, 1, 0, -1, 0};~~

~~2xEHT_LTF_partB =~~

~~{0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0, -1, 0, -1~~

~~0, -1, 0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0, 1~~

~~0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1~~

~~0, 1, 0, -1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, 1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1~~

~~0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, -1, 0, 1~~

~~0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, 1, 0, -1~~

~~0, -1, 0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0, 1~~

~~0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1~~

~~0, 1, 0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1~~

~~0, 1, 0, 1, 0, -1, 0, 1, 0, -1, 0, -1, 0, -1~~

~~0, 1, 0, 1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1~~

~~0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1~~

~~0, -1, 0, -1, 0, 1, 0, -1, 0, 1, 0, 1, 0, 1~~

~~0, -1, 0, 1, 0, 1, 0, 1, 0, -1, 0, 1, 0, -1~~

0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1

0, 1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1

0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1

0, 1, 0, −1, 0, 1, 0, 1, 0, −1, 0, 1, 0, 1

0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1, 0, 1

0, 1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1, 0, −1

0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, 1, 0, −1

0, −1, 0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1

0, −1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1

0, 1, 0, −1, 0, 1, 0, −1, 0, 1, 0, 1, 0, −1

0, 1, 0, 1, 0, −1, 0, −1, 0, 1, 0, −1, 0, −1

0, −1, 0, −1, 0, 1, 0, 1, 0, 1, 0, 1, 0, −1

0, −1, 0, −1, 0, −1, 0, −1, 0, 1, 0, 1, 0, −1

0, 1, 0, −1, 0, 1, 0, −1, 0, −1, 0, −1, 0, 1

0, 1, 0, 1, 0, 1, 0, 1}.

7. An apparatus for transmitting a physical layer protocol data unit, comprising a unit configured to perform the method according to any one of claims 1 and 3 to 6.

8. An apparatus for transmitting a physical layer protocol data unit, comprising a unit configured to perform the method according to any one of claims 2 to 6.

9. An apparatus for transmitting a physical layer protocol data unit, comprising at least one processor, wherein the at least one processor is coupled to a memory, reads computer instructions stored in the memory, and performs, based on the computer instructions, the method according to any one of claims 1 and 3 to 6.

10. An apparatus for transmitting a physical layer protocol data unit, comprising at least one processor, wherein the at least one processor is coupled to a memory, reads computer instructions stored in the memory, and performs, based on the computer instructions, the method according to any one of claims 2 to 6.

11. A computer program, wherein the computer program comprises instructions used to perform the method according to any one of claims 1 and 3 to 6, or comprises instructions used to

~~perform the method according to any one of claims 2 to 6.~~

~~12. A computer-readable storage medium, configured to store a computer program, wherein the computer program comprises instructions used to perform the method according to any one of claims 1 and 3 to 6, or comprises instructions used to perform the method according to any one of claims 2 to 6.~~