WHAT IS CLAIMED IS:

1. A method for data transmission, performed by a packet data convergence protocol (PDCP) entity, wherein the PDCP entity is associated with a radio link control (RLC) entity in an unacknowledged mode (UM), the method comprising:

determining a target working mode of the PDCP entity; and

sending a data packet to a high layer protocol entity in a numbering order based on the target working mode;

in response to the target working mode being a first working mode and before using newly configured PDCP configuration information, sending a first data packet received from a first link to the high layer protocol entity in the numbering order; and

after the newly configured PDCP configuration information is used, sending a second data packet received from a second link to the high layer protocol entity in the numbering order.

2. The method according to claim 1, further comprising:

in response to the target working mode being a second working mode and after the first data packet is sent to the high layer protocol entity in the numbering order, initializing a variable related to a reordering window; and

processing the second data packet by using the newly configured PDCP configuration information based on the reordering window after the initializing.

3. The method according to claim 1, further comprising:

in response to the target working mode being a third working mode, performing an overall reordering on a first data packet and a second data packet based on respective PDCP configuration information, and sending the reordered first data packet and the reordered second data packet to the high layer protocol entity in the numbering order;

wherein the first data packet is a data packet received from a first link, the first data packet corresponds to first PDCP configuration information, and the first PDCP configuration information is previously configured PDCP configuration information; and

the second data packet is a data packet received from a second link, the second data packet corresponds to second PDCP configuration information, and the second PDCP configuration information is newly configured PDCP configuration information.

4. The method according to any one of claims 1 to 3, further comprising:

stopping a timer for reordering.

5. The method according to any one of claims 1 to 3, wherein the first data packet comprises at least one of:

a protocol data unit (PDU) data packet cached in the PDCP entity;

a PDU data packet received by the PDCP entity from the first link before the newly configured PDCP configuration information is used; or

a PDU data packet delivered to the PDCP entity in a reestablishing process of a low layer protocol entity.

6. The method according to any one of claims 1 to 3, further comprising:

processing the first data packet based on the first PDCP configuration information before sending the first data packet to the high layer protocol entity; and

processing the second data packet based on the second PDCP configuration information before sending the second data packet to the high layer protocol entity, wherein the second PDCP configuration information is the newly configured PDCP configuration information.

7. The method according to claim 6, wherein each of the first PDCP configuration information and the second PDCP configuration information comprises at least an encryption algorithm and an encryption key.

8. The method according to claim 7, wherein the first PDCP configuration information and the second PDCP configuration information further comprise at least one of:

a packet header decompression/compression configuration; or

an integrity protection algorithm and an integrity protection key.

9. The method according to claim 8, wherein the packet header decompression/compression configuration comprises at least one of:

a robust header compression (ROHC) /decompression configuration; or an ethernet header compression (EHC) /decompression configuration;

wherein the second PDCP configuration information further comprises indication information for indicating whether a packet header decompression /compression configuration context configured in the first PDCP configuration information continues to be used.

10. The method according to claim 6, wherein the processing comprises a decryption processing and a reordering.

11. The method according to claim 10, wherein the processing further comprises at least one of: a packet header decompression /compression processing or an integrity verification processing.

12. The method according to claim 10, further comprising:

in response to the first data packet being a decrypted data packet, stopping performing the decryption processing on the first data packet.

13. The method according to claim 11, further comprising:

in response to a packet header decompression /compression configuration being configured in the PDCP configuration information, the processing comprising the decompression /compression processing; and

in response to an integrity protection algorithm and an integrity protection key being configured in the PDCP configuration information, the processing comprising the integrity verification processing.

14. The method according to claim 1, further comprising:

in response to the first data packet generating a vacant data packet COUNT in a reordering window, stopping awaiting to receive a data packet corresponding to the vacant data packet COUNT;

in response to a peer PDCP entity numbering the second data packet by using the newly configured PDCP configuration information, enabling a COUNT of the received second data packet and a COUNT of the first data packet to be consecutive COUNTs; and

feeding back data packet COUNT information of the first data packet delivered to the high layer protocol entity to a peer PDCP entity.

15. The method according to claim 1, wherein determining the target working mode of the PDCP entity comprises:

in response to the PDCP entity needing to be reestablished, determining a working mode used by each RLC entity associated with the PDCP entity; and

determining the target working mode of the PDCP entity based on at least one of a working mode used by the RLC entity or a data state feedback indication of the PDCP entity.

16. The method according to claim 15, wherein determining the target working mode of the PDCP entity based on at least one of the working mode used by the RLC entity or the data state feedback indication of the PDCP entity comprises one of:

in response to only the RLC entity in the UM mode being associated with the PDCP entity, determining that the target working mode is a first working mode or a second working mode;

in response to at least one of a plurality of RLC entities associated with the PDCP entity being in an acknowledged mode, determining that the target working mode is a first working mode or a third working mode;

in response to the PDCP entity needing to feed back a data receiving state, determining that the target working mode is a first working mode or a third working mode; or

in response to at least one of a plurality of RLC entities associated with the PDCP entity

receiving data in a multicast mode and at least one of a plurality of RLC entities receiving data in a unicast mode, determining that the target working mode is a first working mode or a third working mode.

17. The method according to claim 16, wherein determining the target working mode of the PDCP entity comprises:

receiving indication information, wherein the indication information carries identity information of the target working mode; and

determining the target working mode based on the indication information.

18. A communication device, comprising:

at least one processor; and

a memory communicatively connected to the at least one processor and stored with instructions executable by the at least one processor,

wherein when the instructions are executed by the at least one processor, the at least one processor is caused to perform the method of any one of claims 1 to 17.

19. A non-transitory computer storage medium storing computer executable instructions, wherein the computer executable instructions are caused to perform the method in any one of claims 1 to 17 when executed by a processor.