

METHOD AND APPARATUS FOR MAPPING VIRTUAL RESOURCES TO PHYSICAL RESOURCES IN A WIRELESS COMMUNICATION SYSTEM

[0001] The present application claims priority to provisional U.S. Application Serial No. 61/072,034, entitled "DOWNLINK DISTRIBUTED TRANSMISSIONS AND OTHER MATTERS," filed March 26, 2008, assigned to the assignee hereof and incorporated herein by reference.

BACKGROUND

I. Field

[0002] The present disclosure relates generally to communication, and more specifically to techniques for determining allocated resources in a wireless communication system.

II. Background

[0003] Wireless communication systems are widely deployed to provide various communication content such as voice, video, packet data, messaging, broadcast, etc. These wireless systems may be multiple-access systems capable of supporting multiple users by sharing the available system resources. Examples of such multiple-access systems include Code Division Multiple Access (CDMA) systems, Time Division Multiple Access (TDMA) systems, Frequency Division Multiple Access (FDMA) systems, Orthogonal FDMA (OFDMA) systems, and Single-Carrier FDMA (SC-FDMA) systems.

[0004] A wireless communication system may include a number of base stations that can support communication for a number of user equipments (UEs). A UE may communicate with a base station via the downlink and uplink. The downlink (or forward link) refers to the communication link from the base station to the UE, and the uplink (or reverse link) refers to the communication link from the UE to the base station.

[0005] A base station may allocate resources to a UE for data transmission on the downlink and/or uplink. It may be desirable to allocate resources such that diversity can

be achieved to obtain good performance while reducing signaling overhead to convey the allocated resources.

SUMMARY

[0006] Techniques for mapping virtual resources to physical resources in a wireless communication system are described herein. Virtual resources in a virtual domain may be allocated to UEs and may be mapped to physical resources usable for communication. In one design, the virtual resources may comprise virtual resource blocks (VRBs), and the physical resources may comprise physical resource blocks (PRBs).

[0007] In an aspect, a virtual resource (e.g., a VRB) may be mapped to a physical resource in a selected subset of physical resources based on a first mapping function. The first mapping function may map contiguous virtual resources to non-contiguous physical resources in the selected subset to achieve diversity as well as other desirable characteristics. The physical resource in the selected subset may then be mapped to an allocated physical resource (e.g., a PRB) among a plurality of available physical resources based on a second mapping function.

[0008] In one design of the first mapping function, an index of the virtual resource may be mapped to a temporary index based on a re-mapping function. The temporary index may then be mapped to an index of the physical resource in the selected subset based on a permutation function. The re-mapping function may be defined to (i) map an input index to two different output indices in two slots to achieve second order diversity when one virtual resource is allocated and (ii) map two consecutive input indices to four different output indices in two slots to achieve fourth order diversity when two virtual resources are allocated. The re-mapping function may also be defined to support efficient allocation of resources, as described below. In one design, the permutation function may map consecutive input indices to permuted output indices to achieve diversity. The permutation function may comprise a bit-reversed row-column interleaver or some other function.

[0009] The selected subset may be one of multiple subsets of physical resources formed with the plurality of available physical resources. The second mapping function

may be applicable for the selected subset. Different second mapping functions may be used for different subsets of physical resources.

[0010] Various aspects and features of the disclosure are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0011] FIG. 1 shows a wireless communication system.
- [0012] FIG. 2 shows an exemplary resource structure.
- [0013] FIG. 3A shows an example of partitioning physical resources into groups.
- [0014] FIG. 3B shows an example of forming subsets of physical resources.
- [0015] FIG. 4 shows an example of mapping a VRB index to a PRB index.
- [0016] FIG. 5 shows a block diagram of a VRB-to-PRB mapping unit.
- [0017] FIG. 6 shows a process for mapping virtual resources to physical resources.
- [0018] FIG. 7 shows an apparatus for mapping virtual resources to physical resources.
- [0019] FIG. 8 shows a block diagram of a base station and a UE.

DETAILED DESCRIPTION

[0020] The techniques described herein may be used for various wireless communication systems such as CDMA, TDMA, FDMA, OFDMA, SC-FDMA and other systems. The terms “system” and “network” are often used interchangeably. A CDMA system may implement a radio technology such as Universal Terrestrial Radio Access (UTRA), cdma2000, etc. UTRA includes Wideband CDMA (WCDMA) and other variants of CDMA. cdma2000 covers IS-2000, IS-95 and IS-856 standards. A TDMA system may implement a radio technology such as Global System for Mobile Communications (GSM). An OFDMA system may implement a radio technology such as Evolved UTRA (E-UTRA), Ultra Mobile Broadband (UMB), IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20, Flash-OFDM®, etc. UTRA and E-UTRA are part of Universal Mobile Telecommunication System (UMTS). 3GPP Long Term Evolution (LTE) and LTE-Advanced (LTE-A) are new releases of UMTS that use E-UTRA, which employs OFDMA on the downlink and SC-FDMA on the uplink. UTRA, E-UTRA, UMTS, LTE, LTE-A and GSM are described in documents from an organization named “3rd Generation Partnership Project” (3GPP). cdma2000 and UMB

are described in documents from an organization named “3rd Generation Partnership Project 2” (3GPP2). The techniques described herein may be used for the systems and radio technologies mentioned above as well as other systems and radio technologies. For clarity, certain aspects of the techniques are described below for LTE, and LTE terminology is used in much of the description below.

[0021] **FIG. 1** shows a wireless communication system 100, which may be an LTE system. System 100 may include a number of evolved Node Bs (eNBs) 110 and other network entities. An eNB may be a station that communicates with the UEs and may also be referred to as a Node B, a base station, an access point, etc. Each eNB 110 provides communication coverage for a particular geographic area and supports communication for the UEs located within the coverage area. UEs 120 may be dispersed throughout the system, and each UE may be stationary or mobile. A UE may also be referred to as a mobile station, a terminal, an access terminal, a subscriber unit, a station, etc. A UE may be a cellular phone, a personal digital assistant (PDA), a wireless modem, a wireless communication device, a handheld device, a laptop computer, a cordless phone, a wireless local loop (WLL) station, etc.

[0022] LTE utilizes orthogonal frequency division multiplexing (OFDM) on the downlink and single-carrier frequency division multiplexing (SC-FDM) on the uplink. OFDM and SC-FDM partition the system bandwidth into multiple (N_{FFT}) orthogonal subcarriers, which are also commonly referred to as tones, bins, etc. Each subcarrier may be modulated with data. In general, modulation symbols are sent in the frequency domain with OFDM and in the time domain with SC-FDM. The spacing between adjacent subcarriers may be fixed, and the total number of subcarriers (N_{FFT}) may be dependent on the system bandwidth. For example, N_{FFT} may be equal to 128, 256, 512, 1024 or 2048 for system bandwidth of 1.25, 2.5, 5, 10 or 20 MHz, respectively.

[0023] **FIG. 2** shows a design of a resource structure 200 that may be used for the downlink or uplink. The transmission timeline may be partitioned into units of subframes. Each subframe may have a predetermined duration, e.g., one millisecond (ms). A subframe may be partitioned into two slots, which may include a first/left slot and a second/right slot. Each slot may include a fixed or configurable number of symbol periods, e.g., six symbol periods for an extended cyclic prefix or seven symbol periods for a normal cyclic prefix.

[0024] N_{RB} resource blocks (RBs) with indices 0 to $N_{RB} - 1$ may be defined in each slot with the N_{FFT} total subcarriers. Each resource block may cover N_{SC} subcarriers (e.g., $N_{SC} = 12$ subcarriers) in one slot. The number of resource blocks in each slot may be dependent on the system bandwidth and may range from 6 to 110. The N_{RB} resource blocks may also be referred to as physical resource blocks (PRBs).

[0025] Virtual resource blocks (VRBs) may also be defined to simplify the allocation of resources. A VRB may have the same dimension as a PRB and may cover N_{SC} subcarriers in one slot in a virtual domain. A VRB may be mapped to a PRB based on a VRB-to-PRB mapping. VRBs may be allocated to the UEs, and transmissions for the UEs may be sent on PRBs to which the allocated VRBs are mapped.

[0026] A UE may be allocated any number of PRBs and any one of the available PRBs. The allocated PRB(s) may be conveyed with a bitmap comprising N_{RB} bits, one bit for each available PRB. Each bitmap bit may be set to '1' to indicate an allocated PRB or to '0' to indicate an un-allocated PRB. However, a large bitmap would be required for a large system bandwidth with a large number of available PRBs.

[0027] To reduce signaling overhead for conveying the allocated resources, the available PRBs may be partitioned into resource block groups (RBGs). Each RBG may include up to P consecutive PRBs, where P may be dependent on the system bandwidth. Table 1 lists the value of P versus system bandwidth for one design.

Table 1 – RBG Size Versus System Bandwidth

System Bandwidth (N_{RB})	RBG Size (P)
≤ 10	1
11 – 26	2
27 – 63	3
63 – 110	4

[0028] N_{RBG} RBGs may be defined with the N_{RB} available PRBs, where N_{RBG} may be given as:

$$N_{RBG} = \lceil N_{RB} / P \rceil, \quad \text{Eq (1)}$$

where $\lceil x \rceil$ denotes a ceiling operator that provides the smallest integer value equal to or greater than x .

[0029] The N_{RBG} RBGs may be divided into P RBG subsets with indices 0 to $P-1$. RBG subset p , for $p = 0, \dots, P-1$, may include every P -th RBG starting with RBG p . The P RBG subsets may or may not include the same number of RBGs, depending on whether $(N_{\text{RBG}} \bmod P) = 0$. The P RBG subsets may or may not include the same number of PRBs, depending on the number of available PRBs and the value of P .

[0030] FIG. 3A shows an example of partitioning the available PRBs into RBGs. In this example, 50 available PRBs with indices $m = 0$ to 49 are partitioned into 17 RBGs with indices 0 to 16. PRB index m may also be referred to as n_{PRB} . Each of the first 16 RBGs includes $P = 3$ contiguous PRBs, and the last RBG includes two remaining PRBs.

[0031] FIG. 3B shows an example of forming three RBG subsets with the 17 RBGs obtained with the 50 available PRBs. RBG subset 0 includes six RBGs with indices 0, 3, 6, 9, 12 and 15 and includes 18 PRBs with indices $m = 0, 1, 2, 9, 10, 11, 18, 19, 20$, etc. RBG subset 1 includes six RBGs with indices 1, 4, 7, 10, 13 and 16 and includes 17 PRBs with indices $m = 3, 4, 5, 12, 13, 14, 21, 22, 23$, etc. RBG subset 2 includes five RBGs with indices 2, 5, 8, 11 and 14 and includes 15 PRBs with indices $m = 6, 7, 8, 15, 16, 17, 24, 25, 26$, etc.

[0032] The system may support multiple resource allocation types, which may include:

- Resource allocation type 0 – allocate an integer number of RBGs,
- Resource allocation type 1 – allocate PRBs within a select RBG subset, and
- Resource allocation type 2 – allocate localized or distributed VRBs.

[0033] For resource allocation type 0, a UE may be allocated any one of the N_{RBG} RBGs. Resource allocation information for the UE may include a bitmap comprising N_{RBG} bits, one bit for each available RBG. Each bitmap bit may be set either to '1' to indicate an allocated RBG or to '0' to indicate an un-allocated RBG. The number of bitmap bits may be reduced by having one bitmap bit for each RBG (instead of each PRB). However, resources are allocated in coarse units of RBG (instead of fine units of PRB).

[0034] For resource allocation type 1, a UE may be allocated any of the PRBs in a selected RBG subset. Resource allocation information for the UE may include (i) an indication of the selected RBG subset and (ii) a bitmap for the PRBs in the selected RBG subset. The bitmap may indicate which PRBs are allocated to the UE.

[0035] For resource allocation type 2, a UE may be allocated a set of contiguous localized or distributed VRBs. A localized VRB with index n_{VRB} may be mapped directly to a PRB with index n_{PRB} , so that $n_{\text{PRB}} = n_{\text{VRB}}$. A distributed VRB (DVRB) with index n_{VRB} may be mapped to a PRB with index n_{PRB} based on a known VRB-to-PRB mapping function $M(\cdot)$, so that $n_{\text{PRB}} = M(n_{\text{VRB}})$. Resource allocation information for the UE may include (i) an indication of whether localized or distributed VRBs are assigned, (ii) an index of the starting VRB allocated to the UE, and (iii) the number of contiguous VRBs allocated to the UE. The terms “contiguous” and “consecutive” may be used interchangeably.

[0036] In an aspect, a VRB-to-PRB mapping may be defined to support distributed VRBs for resource allocation type 2 while being compatible with resource allocation types 0 and 1. In particular, the VRB-to-PRB mapping may map contiguous VRBs to non-contiguous (i.e., permuted, interleaved, or distributed) PRBs within an RBG subset. Restricting the mapping to one RBG subset may allow other RBG subsets to be used for resource allocation type 0 and/or 1.

[0037] In one design, the VRB-to-PRB mapping may comprise (i) a first mapping function $f(\cdot)$ to map contiguous VRBs for one RBG subset to non-contiguous PRBs in the RBG subset and (ii) a second mapping function $s(\cdot)$ to map the PRBs in the RBG subset to all available PRBs. The first mapping function may be defined to support distributed VRBs within one RBG subset. The first mapping function may constrain the mapping of VRBs to PRBs within one RBG subset and may thus allow each RBG subset to be used for any resource allocation type.

[0038] In one design, to simplify the first mapping function, the PRBs in each RBG subset may be aggregated and assigned intermediate indices $k = 0$ to $N_p - 1$, where N_p is the number of PRBs in RBG subset p . N_p may have different values for different RBG subsets and may also be denoted as $N_{\text{RB}}^{\text{RBG}}$. The second mapping function may map

intermediate indices k of the N_p PRBs in RBG subset p to PRB indices m of the N_{RB} available PRBs.

[0039] FIG. 4 shows an example of mapping VRBs for one RBG subset to available PRBs. In this example, 18 VRBs with indices $i = 0$ to 17 may be defined for RBG subset 0. Also, 18 PRBs for RBG subset 0 may be aggregated and assigned intermediate indices $k = 0$ to 17. The first mapping function may map the 18 VRBs to the 18 PRBs in RBG subset 0 and may be implemented as described below.

[0040] The second mapping function may map the 18 PRBs in RBG subset 0 to the 50 available PRBs. Intermediate indices $k = 0, 1$ and 2 for the three PRBs in the first RBG may be mapped to PRB indices $m = 0, 1$ and 2. Intermediate indices $k = 3, 4$ and 5 for the three PRBs in the next RBG may be mapped to PRB indices $m = 9, 10$ and 11. Intermediate indices for the remaining PRBs in RBG subset 0 may be mapped to PRB indices for the available PRBs as shown in FIG. 4. FIG. 4 shows an example of the second mapping function for RBG subset 0. Different second mapping functions may be defined for different RBG subsets. The second mapping functions for all P RBG subsets may be implemented in an efficient manner due to the structured manner in which the RBGs and the RBG subsets are defined.

[0041] In one design, the first mapping function may map contiguous VRBs for one RBG subset to non-contiguous PRBs in the RBG subset. In general, N_p VRBs may be available for RBG subset p and may be assigned indices $i = 0$ to $N_p - 1$. VRB index i may also be referred to as n_{VRB} . N_p PRBs may also be included in RBG subset p and may be assigned indices $k = 0$ to $N_p - 1$. The first mapping function $f(\)$ may map VRB index i to intermediate index k , so that $k = f(i)$.

[0042] The first mapping function $f(\)$ may be implemented in various manners. In one design, the first mapping function may comprise a re-mapping function $r(\)$ and a permutation function $\sigma(\)$. The re-mapping function may map input indices to output indices to obtain certain desired characteristics, as described below. The permutation function may map contiguous input indices to permuted output indices to achieve diversity. In one design that is described below, the re-mapping function may be followed by the permutation function. In this design, the re-mapping function may map VRB index i to a temporary index j , or $j = r(i)$, and the permutation function may map

temporary index j to intermediate index k , or $k = \sigma(j)$. In another design that is not described below, the permutation function may be followed by the re-mapping function. In this design, the permutation function may map VRB index i to temporary index j , and the re-mapping function may map temporary index j to intermediate index k .

[0043] In one design, the re-mapping function $r(\cdot)$ may map input indices to output indices to obtain certain desired characteristics. The re-mapping function may map an input index i to two different output indices j in two slots of a subframe to obtain second order diversity when one VRB is allocated. The re-mapping function may map two input indices i to four different output indices j in two slots of a subframe to obtain fourth order diversity when two VRBs are allocated. In one design, the re-mapping function may be complementary or symmetric, so that $j = r(i)$ for the first slot and $j = r(N_p - i - 1)$ for the second slot. In this design, the mapping of smallest to largest input indices i to output indices j in the first slot matches the mapping of largest to smallest input indices i to output indices j in the second slot.

[0044] Table 2 shows a design of the re-mapping function for one RBG subset. In the first slot, the re-mapping function may map input index $i = 0$ to output index $j = 0$, input index $N_p - 1$ to output index 1, input index 1 to output index 2, input index $N_p - 2$ to output index 3, and so on. In the second slot, the re-mapping function may map input index $i = N_p - 1$ to output index $j = 0$, input index 0 to output index 1, input index $N_p - 2$ to output index 2, input index 1 to output index 3, and so on. As shown in Table 2, a given VRB may be mapped to two PRBs in an RBG subset in two slots and may then achieve second order diversity. A pair of VRBs may be mapped to four different PRBs in an RBG subset in two slots and may then achieve fourth order diversity. For example, VRBs 0 and 1 may be mapped to four different PRBs $\sigma(0)$, $\sigma(1)$, $\sigma(2)$ and $\sigma(3)$ in two slots.

Table 2 – Re-mapping Function

First Slot			Second Slot		
VRB Index i	Temporary Index j	Intermediate Index k	VRB Index i	Temporary Index j	Intermediate Index k
0	0	$\sigma(0)$	0	1	$\sigma(1)$
1	2	$\sigma(2)$	1	3	$\sigma(3)$

2	4	$\sigma(4)$	2	5	$\sigma(5)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N_p-3	5	$\sigma(5)$	N_p-3	4	$\sigma(4)$
N_p-2	3	$\sigma(3)$	N_p-2	2	$\sigma(2)$
N_p-1	1	$\sigma(1)$	N_p-1	0	$\sigma(0)$

[0045] As shown in Table 2, the re-mapping function may map indices in complementary order for the first and second slots. This may result in a natural pairing of indices (and hence PRBs) for two slots and may preserve as much PRBs as possible for resource allocation type 0. In one design, VRB allocation may be performed by allocating a set of contiguous VRBs from the top, then from the bottom, then from the top, and so on, moving from the two ends toward the middle. For example, a first UE may be allocated VRBs 0 and 1 and may use PRBs $\sigma(0)$ and $\sigma(2)$ in the first slot and PRBs $\sigma(2)$ and $\sigma(3)$ in the second slots. A second UE may be allocated VRBs N_p-2 and N_p-1 and may use PRBs $\sigma(1)$ and $\sigma(3)$ in the first slot and PRBs $\sigma(0)$ and $\sigma(2)$ in the second slots. These two UEs may each achieve fourth order diversity using four PRBs $\sigma(0)$, $\sigma(1)$, $\sigma(2)$ and $\sigma(3)$ in two slots. The remaining PRBs may be allocated to other UEs using any resource allocation type. Allocation of VRBs from the two ends and gradually moving toward the middle, using the re-mapping function shown in Table 2, may efficiently conserve PRBs. The re-mapping function may also map input indices i to output indices j in other manners.

[0046] In one design, the permutation function $\sigma(\)$ may be based on a bit-reversed row-column interleaver. For the interleaver, a table with C column and R rows may be defined, where C may be dependent on the system bandwidth and may be defined as shown in Table 3.

Table 3 – Number of Columns Versus System Bandwidth

System Bandwidth (N_{RB})	Number of Columns (C)
≤ 63	4
64 – 110	8

[0047] The number of rows (R) may be computed based on the number of columns (C), as follows:

$$R = \lceil N_p / C \rceil . \quad \text{Eq (2)}$$

[0048] The operation of the bit-reversed row-column interleaver may be as follows:

1. Create a rectangular table of R rows and C columns.
2. Write numbers $i = 0$ to $N_p - 1$ sequentially column-wise from top to bottom and from left to right. Insert filler elements in $R \cdot C - N_p$ locations in the last row of the table, e.g., starting from the right.
3. Perform bit-reversed interleaving of the C columns. The C columns may be assigned indices 0 to $C - 1$. The index of each column may have a binary representation of $[b_0 b_1 \dots b_c]$, where b_j denotes the j -th bit of the column index. A column with index $[b_0 b_1 \dots b_c]$ may be swapped with a column with index $[b_c \dots b_1 b_0]$.
4. Read out the elements of the table sequentially row-wise from left to right and from top to bottom. Skip any filler elements.

[0049] The sequence of elements read from the table may be denoted as $\sigma(0)$, $\sigma(1)$, $\sigma(2)$, ..., $\sigma(N_p - 1)$. These elements represent temporary indices to which the VRB indices are mapped. In particular, the i -th index written to the table is mapped to the i -th element read from the table, so that VRB index i is mapped to intermediate index $\sigma(i)$. Since elements $\sigma(i)$, $\sigma(i + 1)$, $\sigma(i + 2)$ and $\sigma(i + 3)$ are consecutively read from the table, fourth order diversity may be achieved for an allocation of two contiguous VRBs.

[0050] For the example shown in FIG. 4 with 18 VRBs for one RBG subset, a table with five rows and four columns may be defined. Numbers 0 to 4 may be written from top to bottom in column 0, numbers 5 to 9 may be written in column 1, numbers 10 to 13 may be written in column 2, and numbers 14 to 17 may be written in column 3, as shown in the first four columns of Table 4. Columns 1 and 2 may be swapped due to bit-reversed interleaving, and the table after column permutation may be as shown in the last four columns of Table 4. The 18 elements of the table may be read out row-wise

from left to right and from top to bottom to obtain a sequence of 0, 10, 5, 14, 1, 11, 6, 15, 2, 12, 7, 16, 3, 13, 8, 17, 4 and 9 for $\sigma(0)$ to $\sigma(17)$.

Table 4 – Bit-Reversed Row-Column Interleaver

	Column				Column			
Row	0	1	2	3	0	1	2	3
0	0	5	10	14	0	10	5	14
1	1	6	11	15	1	11	6	15
2	2	7	12	16	2	12	7	16
3	3	8	13	17	3	13	8	17
4	4	9			4		9	

[0051] Table 5 shows an example of the first mapping function $k = f(i)$ for 18 VRBs in one RBG subset for the first slot. The first mapping function comprises the re-mapping function $j = r(i)$ and the permutation function $k = \sigma(j)$. Temporary index j in the second and fifth columns of Table 5 may be obtained by applying the re-mapping function shown in Table 2 on VRB index i . Intermediate index k in the third and last columns of Table 5 may be obtained by applying the permutation function described above on temporary index j .

Table 5 – VRB-to-PRB Mapping for One RBG Subset

VRB Index i	Temporary Index j	Intermediate Index k	VRB Index i	Temporary Index j	Intermediate Index k
0	0	0	9	17	9
1	2	5	10	15	17
2	4	1	11	13	13
3	6	6	12	11	16
4	8	2	13	9	12
5	10	7	14	7	15
6	12	3	15	5	11
7	14	8	16	3	14
8	16	4	17	1	10

[0052] Table 5 shows one design of the re-mapping function and one design of the permutation function for a specific case with 18 VRBs. The re-mapping function may also be implemented in other manners. For example, the re-mapping function may map input indices 0, 1, 2, 3, etc., to output indices 0, $N_p/2$, 1, $N_p/2+1$, etc. in the first slot and to output indices $N_p/4$, $3N_p/4$, $N_p/4+1$, $3N_p/4+1$, etc. in the second slot. The re-mapping function may also map input indices to output indices in other manners.

[0053] The permutation function may be implemented with a bit-reversed row-column interleaver, as described above. In general, the table used for the bit-reversed row-column interleaver may include any number of columns that is a power of two and any number of rows. The permutation function may also be implemented with a bit-reversed interleaver, a pseudo-random interleaver, or some other permutation function that maps consecutive input indices to non-consecutive (i.e., permuted) output indices.

[0054] FIG. 5 shows a block diagram of a design of a VRB-to-PRB mapping unit 500. A VRB index i for a VRB allocated to a UE may be provided to both (i) a block 510a for the first mapping function $f(\cdot)$ for the first slot of a subframe and (ii) a block 510b for the first mapping function for the second slot of the subframe.

[0055] Within block 510a, a block 512a may implement the re-mapping function $r(\cdot)$ for the first slot and may receive VRB index i and provide temporary index j_1 for the first slot. A block 514a may implement the permutation function $\sigma(\cdot)$ and may receive temporary index j_1 and provide intermediate index k_1 for the first slot. A block 520a may implement the second mapping function $s(\cdot)$ and may receive intermediate index k_1 and provide PRB index m_1 for the first slot.

[0056] Within block 510b, a block 512b may implement the re-mapping function for the second slot and may receive VRB index i and provide temporary index j_2 for the second slot. A block 514b may implement the permutation function and may receive temporary index j_2 and provide intermediate index k_2 for the second slot. A block 520b may implement the second mapping function and may receive intermediate index k_2 and provide PRB index m_2 for the second slot.

[0057] The re-mapping function may map contiguous VRB indices to temporary indices in a complementary manner for the first and second slots, e.g., as shown in Table 2. The same permutation function and the same second mapping function may be used for both the first and second slots. The permutation function may be implemented

with the bit-reversed row-column interleaver, as described above, or with some other design. The second mapping function may be dependent on the selected RBG subset, as described above.

[0058] The mapping techniques described herein may provide certain advantages. First, the permutation function may provide diversity while keeping allocation for a given UE to the same RBG subset. This may allow each RBG subset to be used for any resource allocation type and may further allow for mixing of different resource allocation types for different RBG subsets. Second, the re-mapping function may ensure pairing of PRBs in the two slots of a subframe. The re-mapping function and the permutation function may operate on indices with a range of 0 to N_p-1 (instead of 0 to $N_{RB}-1$), which may simplify the design of these functions. The second mapping function may map PRBs in one RBG subset to available PRBs and may be easily implemented.

[0059] FIG. 6 shows a design of a process 600 for mapping virtual resources to physical resources. Process 600 may be performed by a UE, an eNB/base station, or some other entity. A virtual resource may be mapped to a physical resource in a selected subset of physical resources (block 612). The selected subset may be one of multiple subsets of physical resources formed with a plurality of available physical resources. Contiguous virtual resources may be mapped to non-contiguous physical resources in the selected subset. The physical resource in the selected subset may be mapped to an allocated physical resource among the plurality of available physical resources (block 614). The allocated physical resource may be used for communication (e.g., to send or receive data) in a first slot (block 616).

[0060] The virtual resource may also be mapped to a second physical resource in the selected subset (block 618). The second physical resource in the selected subset may be mapped to a second allocated physical resource among the plurality of available physical resources (block 620). The second allocated physical resource may be used for communication in a second slot (block 622).

[0061] In one design, K virtual resources may be mapped to K physical resources in the selected subset based on a first mapping function. The K physical resources in the selected subset may be mapped to N available physical resources based on a second mapping function. K may be greater than one and may correspond to N_p above. N may be greater than K and may correspond to N_{RB} above.

[0062] In one design of the first mapping function, an index of the virtual resource may be mapped to a temporary index based on a re-mapping function. The temporary index may be mapped to an index of the physical resource in the selected subset based on a permutation function. The order of the re-mapping function and the permutation function may also be swapped.

[0063] In one design, the re-mapping function may map the smallest to largest input indices to output indices based on a predetermined mapping in the first slot. The re-mapping function may map the largest to smallest input indices to output indices based on the predetermined mapping in the second slot. The re-mapping function may map an input index to two different output indices for the first and second slots to achieve second order diversity when one virtual resource is allocated. The re-mapping function may also map two consecutive input indices to four different output indices in the first and second slots to achieve fourth order diversity when two virtual resources are allocated.

[0064] In one design, the permutation function may map consecutive input indices to permuted output indices to achieve diversity. The permutation function may comprise a bit-reversed row-column interleaver or some other function.

[0065] In one design, the second mapping function may map physical resources in the selected subset to the plurality of available physical resources and may be applicable for the selected subset. Different second mapping functions may be applicable for different subsets of physical resources.

[0066] In one design, the virtual resource may comprise a VRB, and the allocated physical resource may comprise a PRB. The virtual resource and the allocated physical resource may also comprise other types of resources. In one design, the plurality of available physical resources may be partitioned into multiple groups, with each group including up to P contiguous physical resources, e.g., as shown in FIG. 3A. P may be one or greater and may be dependent on the number of available physical resources. P subsets of physical resources may be formed, with each subset including every P -th group starting with an initial group for the subset. The multiple subsets of physical resources may also be formed in other manners.

[0067] In one design, a set of virtual resources may be associated with the selected subset of physical resources. Virtual resources in the set may be allocated to users or

UEs by alternating between two ends of the set and moving from the two ends toward the middle of the set.

[0068] In one design, the physical resources in each subset may be allocated to users based on at least one resource allocation type. The at least one resource allocation type may include (i) a first resource allocation type (type 0) used to allocate one or more groups of physical resources in a subset of physical resources, (ii) a second resource allocation type (type 1) used to allocate one or more physical resources in a subset of physical resources, and/or (iii) a third resource allocation type (type 2) used to allocate contiguous virtual resources mapped to non-contiguous physical resources in a subset of physical resources. The physical resources in each subset may also be allocated to users in other manners.

[0069] FIG. 7 shows a design of an apparatus 700 for mapping resources. Apparatus 700 includes a module 712 to map a virtual resource to a physical resource in a selected subset of physical resources for a first slot, a module 714 to map the physical resource in the selected subset to an allocated physical resource among a plurality of available physical resources for the first slot, a module 716 to use the allocated physical resource for communication in the first slot, a module 718 to map the virtual resource to a second physical resource in the selected subset for a second slot, a module 720 to map the second physical resource in the selected subset to a second allocated physical resource among the plurality of available physical resources for the second slot, and a module 722 to use the second allocated physical resource for communication in the second slot.

[0070] The modules in FIG. 7 may comprise processors, electronics devices, hardware devices, electronics components, logical circuits, memories, software codes, firmware codes, etc., or any combination thereof.

[0071] FIG. 8 shows a block diagram of a design of an eNB/base station 110 and a UE 120, which may be one of the eNBs and one of the UEs in FIG. 1. eNB 110 may be equipped with T antennas 834a through 834t, and UE 120 may be equipped with R antennas 852a through 852r, where in general $T \geq 1$ and $R \geq 1$.

[0072] At eNB 110, a transmit processor 820 may receive data for one or more UEs from a data source 812, process (e.g., encode, interleave, and modulate) the data for each UE based on one or more modulation and coding schemes for that UE, and provide data symbols for all UEs. Transmit processor 820 may also process control information

(e.g., scheduling information for allocated resources) from a controller/ processor 840 and provide control symbols. A transmit (TX) multiple-input multiple-output (MIMO) processor 830 may multiplex the data symbols, the control symbols, and/or pilot symbols. TX MIMO processor 830 may perform spatial processing (e.g., precoding) on the multiplexed symbols, if applicable, and provide T output symbol streams to T modulators (MODs) 832a through 832t. Each modulator 832 may process a respective output symbol stream (e.g., for OFDM) to obtain an output sample stream. Each modulator 832 may further process (e.g., convert to analog, amplify, filter, and upconvert) the output sample stream to obtain a downlink signal. T downlink signals from modulators 832a through 832t may be transmitted via T antennas 834a through 834t, respectively.

[0073] At UE 120, antennas 852a through 852r may receive the downlink signals from eNB 110 and provide received signals to demodulators (DEMODs) 854a through 854r, respectively. Each demodulator 854 may condition (e.g., filter, amplify, downconvert, and digitize) a respective received signal to obtain received samples. Each demodulator 854 may further process the received samples (e.g., for OFDM) to obtain received symbols. A MIMO detector 856 may obtain received symbols from all R demodulators 854a through 854r, perform MIMO detection on the received symbols if applicable, and provide detected symbols. A receive processor 858 may process (e.g., demodulate, deinterleave, and decode) the detected symbols, provide decoded control information to a controller/processor 880, and provide decoded data for UE 120 to a data sink 860.

[0074] On the uplink, at UE 120, data from a data source 862 and control information from controller/processor 880 may be processed by a transmit processor 864, precoded by a TX MIMO processor 866 if applicable, conditioned by modulators 854a through 854r, and transmitted to eNB 110. At eNB 110, the uplink signals from UE 120 may be received by antennas 834, conditioned by demodulators 832, processed by a MIMO detector 836 if applicable, and further processed by a receive processor 838 to obtain the data and control information transmitted by UE 120.

[0075] Controllers/processors 840 and 880 may direct the operation at eNB 110 and UE 120, respectively. Processor 880 and/or other processors and modules at UE 120 may implement the VRB-to-PRB mapping shown in FIGS. 4 and 5 and may perform or direct process 600 in FIG. 6 and/or other processes for the techniques described herein.

Processor 840 and/or other processors and modules at eNB 110 may also implement the VRB-to-PRB mapping shown in FIGS. 4 and 5 and may also perform or direct process 600 in FIG. 6 and/or other processes for the techniques described herein. Memories 842 and 882 may store data and program codes for eNB 110 and UE 120, respectively. A scheduler 844 may schedule UEs for downlink and/or uplink transmission and may provide allocations of resources for the scheduled UEs.

[0076] Those of skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0077] Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the disclosure herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0078] The various illustrative logical blocks, modules, and circuits described in connection with the disclosure herein may be implemented or performed with a general-purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of

a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0079] The steps of a method or algorithm described in connection with the disclosure herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0080] In one or more exemplary designs, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage media may be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code means in the form of instructions or data structures and that can be accessed by a general-purpose or special-purpose computer, or a general-purpose or special-purpose processor. Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where *disks* usually

reproduce data magnetically, while *discs* reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0081] The previous description of the disclosure is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to the disclosure will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other variations without departing from the spirit or scope of the disclosure. Thus, the disclosure is not intended to be limited to the examples and designs described herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[0082] **WHAT IS CLAIMED IS:**

CLAIMS

1. A method performed by a wireless communication apparatus, comprising:

Mapping, by a processor, a virtual resource to a physical resource in a selected subset of physical resources by mapping an index of the virtual resource to a temporary index based on a re-mapping function and mapping the temporary index to an index of the physical resource in the selected subset based on a permutation function, wherein the selected subset is one of multiple subsets of physical resources formed with a plurality of available physical resources, wherein the re-mapping function maps smallest to largest input indices to output indices based on a predetermined mapping in a first slot and maps largest to smallest input indices to output indices based on the predetermined mapping in a second slot, and wherein contiguous virtual resources are mapped to non-contiguous physical resources in the selected subset;

mapping the physical resource in the selected subset to an allocated physical resource among the plurality of available physical resources; and

using the allocated physical resource for communication.

2. The method of claim 1, wherein the selected subset includes K physical resources, wherein K virtual resources are mapped to the K physical resources in the selected subset based on a first mapping function, and wherein the K physical resources in the selected subset are mapped to N available physical resources based on a second mapping function, where K is greater than one, and N is greater than K .

3. The method of claim 1, wherein the mapping the virtual resource comprises mapping the virtual resource to the physical resource in the selected subset based on a permutation function, the permutation function mapping consecutive input indices to permuted output indices to achieve diversity.

4. The method of claim 3, wherein the permutation function comprises a bit-reversed row-column interleaver.

5. The method of claim 1, wherein the allocated physical resource is used for communication in a first slot, the method further comprising:

mapping the virtual resource to a second physical resource in the selected subset;
mapping the second physical resource in the selected subset to a second allocated physical resource among the plurality of available physical resources; and
using the second allocated physical resource for communication in a second slot.

6. The method of claim 1, wherein the mapping the physical resource in the selected subset comprises mapping the physical resource in the selected subset to the allocated physical resource based on a mapping function applicable for the selected subset, wherein different mapping functions are applicable for the multiple subsets of physical resources.

7. The method of claim 1, wherein the virtual resource comprises a virtual resource block (VRB) and the allocated physical resource comprises a physical resource block (PRB).

8. The method of claim 1, wherein the plurality of available physical resources are partitioned into multiple groups, each group including up to P contiguous physical resources, where P is one or greater, and wherein P subsets of physical resources are formed, with each subset including every P -th group starting with an initial group for the subset.

9. The method of claim 1, wherein a set of virtual resources is associated with the selected subset of physical resources, the method further comprising:

allocating virtual resources in the set to users by alternating between two ends of the set and moving from the two ends toward middle of the set.

10. The method of claim 1, further comprising:

allocating physical resources in each of the multiple subsets based on at least one resource allocation type.

11. The method of claim 10, wherein the at least one resource allocation type comprises at least one of a first resource allocation type used to allocate one or more groups of physical

resources in a subset of physical resources, a second resource allocation type used to allocate one or more physical resources in a subset of physical resources, or a third resource allocation type used to allocate contiguous virtual resources mapped to non-contiguous physical resources in a subset of physical resources.

12. The method of claim 1, wherein the using the allocated physical resource comprises sending or receiving data on the allocated physical resource.

13. An apparatus for wireless communication, comprising:

at least one processor configured to map a virtual resource to a physical resource in a selected subset of physical resources by mapping an index of the virtual resource to a temporary index based on a re-mapping function and mapping the temporary index to an index of the physical resource in the selected subset based on a permutation function, to map the physical resource in the selected subset to an allocated physical resource among a plurality of available physical resources, and to use the allocated physical resource for communication, wherein the selected subset is one of multiple subsets of physical resources formed with the plurality of available physical resources, wherein the re-mapping function maps smallest to largest input indices to output indices based on a predetermined mapping in a first slot and maps largest to smallest input indices to output indices based on the predetermined mapping in a second slot, and wherein contiguous virtual resources are mapped to non-contiguous physical resources in the selected subset.

14. The apparatus of claim 13, wherein the at least one processor is configured to map K virtual resources to K physical resources in the selected subset based on a first mapping function, and to map the K physical resources in the selected subset to N available physical resources based on a second mapping function, where K is greater than one, and N is greater than K .

15. The apparatus of claim 13, wherein the at least one processor is configured to map the virtual resource to the physical resource in the selected subset based on a permutation function, the permutation function mapping consecutive input indices to permuted output indices to achieve diversity.

16. The apparatus of claim 13, wherein the at least one processor is configured to use the allocated physical resource for communication in a first slot, to map the virtual resource to a second physical resource in the selected subset, to map the second physical resource in the selected subset to a second allocated physical resource among the plurality of available physical resources, and to use the second allocated physical resource for communication in a second slot.

17. An apparatus for wireless communication, comprising:

means for mapping a virtual resource to a physical resource in a selected subset of physical resources by mapping an index of the virtual resource to a temporary index based on a re-mapping function and mapping the temporary index to an index of the physical resource in the selected subset based on a permutation function, wherein the selected subset is one of multiple subsets of physical resources formed with a plurality of available physical resources, wherein the re-mapping function maps smallest to largest input indices to output indices based on a predetermined mapping in a first slot and maps largest to smallest input indices to output indices based on the predetermined mapping in a second slot, and wherein contiguous virtual resources are mapped to non-contiguous physical resources in the selected subset;

means for mapping the physical resource in the selected subset to an allocated physical resource among the plurality of available physical resources; and

means for using the allocated physical resource for communication.

18. The apparatus of claim 17, wherein the selected subset includes K physical resources, wherein K virtual resources are mapped to the K physical resources in the selected subset based on a first mapping function, and wherein the K physical resources in the selected subset are mapped to N available physical resources based on a second mapping function, where K is greater than one, and N is greater than K .

19. The apparatus of claim 17, wherein the means for mapping the virtual resource comprises means for mapping the virtual resource to the physical resource in the selected subset based on a permutation function, the permutation function mapping consecutive input indices to permuted output indices to achieve diversity.

20. The apparatus of claim 17, wherein the allocated physical resource is used for communication in a first slot, the apparatus further comprising:

means for mapping the virtual resource to a second physical resource in the selected subset;

means for mapping the second physical resource in the selected subset to a second allocated physical resource among the plurality of available physical resources; and

means for using the second allocated physical resource for communication in a second slot.

21. A computer program product, comprising:

a non-transitory computer-readable medium comprising:

code for causing at least one computer to map a virtual resource to a physical resource in a selected subset of physical resources by mapping an index of the virtual resource to a temporary index based on a re-mapping function and mapping the temporary index to an index of the physical resource in the selected subset based on a permutation function, wherein the selected subset is one of multiple subsets of physical resources formed with a plurality of available physical resources, wherein the re-mapping function maps smallest to largest input indices to output indices based on a predetermined mapping in a first slot and maps largest to smallest input indices to output indices based on the predetermined mapping in a second slot, and wherein contiguous virtual resources are mapped to non-contiguous physical resources in the selected subset,

code for causing at least one computer to map the physical resource in the selected subset to an allocated physical resource among the plurality of available physical resources, and

code for causing at least one computer to use the allocated physical resource for communication.