TSG-RAN Working Group 1 meeting #6
Espoo, Finland
July 13-16, 1999

TSGR1#6(99)884

Agenda Item:

Source: Ericsson

Title: New downlink scrambling code grouping scheme for UTRA/FDD, revised

Document for: Decision

1. Introduction

In [1], a new downlink scrambling code grouping scheme was proposed. The proposed scheme was shown to reduce the complexity and hardware requirements for cell search, at the expense of a small increase in average cell search time. However, a few issues related to this proposal were raised recently. These include peak processing power and cell search performance at individual stages. In this document, these issues are addressed. First, in addition to the originally proposed scheme (with 256 scrambling code groups), two new alternatives are proposed. The new proposed alternatives are based on increasing the number of scrambling code groups to 64, and 128, respectively. These alternatives are favorable compared to the current scheme (with 32 scrambling code groups) as far as the complexity of cell search is concerned. The complexity of the current scheme and the proposed schemes are analyzed. In our revised analysis, the operations required for memory access are included. In addition, the storage requirement is also analyzed. Second, the concern of peak processing power is addressed. It was found that peak memory access occurs during the Reed-Solomon (RS) decoding process of stage 2. To lessen the peak memory access rate, extra time slots can be allocated for RS decoding. Finally, the simulation results for stages 2 and 3 are presented. In the analysis the comparison is done for 4.096 Mcps, however in the text proposal the change to the harmonized scheme is made.

2. Current scheme and the proposed scheme

The cell search procedure according to [2] and [3] consists of three stages: (1) matched filtering (slot boundary detection), (2) RS code decoding (scrambling code group and frame timing identification), and (3) scrambling code identification. In the first stage the received signal is passed through a matched filter, matching to the 256-chip primary synchronization code (PSC). By detecting the peaks at the output of the PSC matched filter, the slot boundary can be found. In the second stage, the received signal is correlated against 17 Secondary Synchronization Codes (SSC). The SSC carries different symbols in different time slots according to a (16,3) RS code used for frame synchronization and scrambling code group identification. By decoding the RS code, frame boundary and scrambling code identity can be found. In the third stage, the received signal is correlated against all possible scrambling codes in the identified code group (CG). According to the current scheme, the number of possible scrambling codes in the identified CG is 16 for initial cell search, or at least 1-4 for target cell search, if a good system code planning is used.

Originally, the complexity of cell search was dominated by the first stage. However, a new PSC based on generalized hierarchical sequence was introduced and adopted recently [5]. With this new PSC and a proposed pruned efficient Golay correlator [4], the complexity of stage 1 is significantly reduced. This leaves stage 3 as the most complex stage. As mentioned earlier, the number of scrambling codes in the identified CG can be up to 16. This requires the MS having 16 correlators dedicated for scrambling code correlations during initial cell search. If an MS is not equipped with 16 correlators, scrambling code correlations can not be performed in parallel, giving rise to an increase in synchronization time. Furthermore, it will be shown in the next section that the current scrambling code grouping scheme requires a rather high total complexity.

To lessen the requirement of having 16 correlators to carry out scrambling code correlations in parallel and reduce the complexity of initial cell search, we propose that the number of CG to be increased. This reduces the number of scrambling codes in a CG, making it easy to perform parallel scrambling code correlations within the identified CG. In this document, we propose three alternatives to achieve this. The proposed schemes use either 256, 128, or 64 codewords from a Comma-Free RS code. This can be achieved by using a (16,3) RS code over GF(17), which is the same code as used in [2]. However, in [2] only 32 comma-free codewords are used. The codebook used in [2] and the ones used in the proposed scheme all have minimum symbol distance 14. It can be shown that with the proposed schemes, the complexity of initial cell search can be reduced by 39% to 24% compared to the current scheme.

In cell reselection and target cell search scenarios the MS knows the different scrambling codes of the neighboring cells through signalling. Due to fact that the number of CG is larger in the proposed schemes it is possible for the MS to uniquely identify the scrambling code by identifying the CG. Thus the stage 3 of the cell search is not needed during handoff scenarios according to the proposed schemes.

In the current scheme the code planning plays an important role when it comes to complexity for the MS. This is because the CG can be assigned in a clustered or a distributed manner according to [4].

The scheme using 256 CG's is the preferred scheme among the three proposed schemes. It achieves the most complexity reduction and requires only 2 scrambling code correlators during stage 3 of initial cell search. In addition, the number of cell specific preamble spreading codes for random access is currently also equal to 256. By having the same number of codes both for number of RACH identities and the number of scrambling code groups there is also an option for the operators to tie the preamble sequence with a specific scrambling code group. This is of course up to the operators how to allocate the codes.

3. Complexity and Memory Requirement

In this section, the complexity in each of the stages is analyzed. In addition, the memory requirement is also addressed.

First define the following parameters:

- M: the number of scrambling code groups,
- L: the number of scrambling codes in a code group,
- N_1: synchronization time (in frames) for stage 1,
- N_2: synchronization time (in frames) for stage 2,
- N_3: synchronization time (in frames) for stage 3,
- C_1: complexity of stage 1 (number of operations), excluding memory access,
- C_2: complexity of stage 2 (number of operations), excluding memory access,
- C_3: complexity of stage 3 (number of operations) , excluding memory access,
- Y : number of memory access operations.

3.1 Complexity (Excluding Memory Access)

The complexity of stage 1 according to the PSC based on Generalized Hierarchical Sequence is 13*2 operations per matched filter output. Assuming one sample per chip for stage 1,

$$\begin{aligned} C_1 &= 13*2*40960*N_1 \\ &= 1064960\,N_1 . \end{aligned} \qquad (1)$$

The complexity of the stage 2 is

$$C_2 = (32*5+32*7)*2*16\,N_2 + 17*16*3*N_2 + 16*16\,M + 16M \qquad (2)$$
$$= 13104\,N_2 + 272M$$

In (2), the term "(32*5+32*7)" is for correlating with 17 Second Search Codes (SSC), utilizing Fast Walsh Transform (FWT). This term is multiplied by 2, for SSC correlations on I and Q channels, and then by 16 N_2, for there are 16 N_2 SSC symbols in N_2 frames. The term "17*16*3* N_2 " is for coherent demodulation of SSC symbols, assuming three operations for calculating the real part of a complex product. The term "16*16*M* " is for calculating the metric for decoding the Comma-Free RS code. There are 16*M* hypotheses (*M* code groups * 16 shifts) and each hypothesis corresponds to a codeword of length 16. Finally, the term "16*M*" is for finding the maximum metric among all 16*M* hypotheses.

The complexity of stage 3 is given by

$$C_3 = 256*9*L*16*N_3*2 \qquad (3)$$
$$= 73728L\,N_3\,.$$

In (3), the term "256*9" is due to there are 256*9 chips in a slot scrambled by the scrambling code. This term is first multiplied by *L*, for there are *L* scrambling codes to be correlated against, and then by 16 N_3, for there are 16 N_3 slots for stage 3, and finally by 2, for accounting for scrambling code correlations on both I and Q channels.

3.2 Memory Access

In stage 1, each PSC matched filter output needs to be accumulated. This requires one memory read and write per chip. Thus in a frame there are 81920 memory accesses. In stage 2, the SSC correlations need to be accumulated. This requires 17 memory read and write per slot. In a frame, there are 544 memory accesses during SSC correlations. During RS decoding, memory access of *M**16*16+16*M*=272*M* times is required. In stage 3, to accumulate the scrambling code correlation metrics, 9*16**L**2 memory accesses are needed in one frame, assuming a despreading factor of 256. Thus the total amount of memory access is given by

$$Y = 81920\,N_1 + 544\,N_2 + 272\,M + 288\,L\,N_3\,. \qquad (4)$$

From the above analysis, the complexity of cell search with $N_1 = N_2 = N_3 = 1$ and various choices of *M* and *L* is listed in Table 1. The reduction complexity compared to the current scheme is also listed in Table 1.

Table 1: Complexity of the current scheme and proposed schemes. (with memory access operations included)

scheme	*M*	*L*	complexity	complexity reduction
current scheme	32	16	2362192	-
1 (preferred scheme)	256	2	1447824	39%
2	128	4	1526224	35%
3	64	8	1787472	24%

It can be seen that complexity reduction of 39% to 24% can be achieved by the proposed schemes.

3.3 Storage Requirement

The storage requirement for the current scheme and all the proposed schemes are roughly the same, if RS decoding is performed after finishing SSC correlations. In stage 1, 2560 elements are needed for storing the accumulated metrics corresponding to 2560 bins in one timeslot interval. In stage 2, 17 SSC correlations have to be stored for each time slot, resulting in 17*16=272 memory elements. In addition, two extra memory elements are required to store the largest metric and index during the RS decoding process. The RS code words can be generated on the fly or pre-stored in a code word table. If it is pre-stored in a table, the code word table requires M*16 memory elements. In stage 3, L memory elements are needed to store the metrics corresponding to the L scrambling code correlations. Similar to stage 2, two extra elements are needed for storing the largest metric and index during the process of searching for the largest metric and candidate. Table 2 lists the memory requirement when one sample per chip is used throughout the cell search process. It can be seen that the proposed schemes require a lot more memory elements, mainly for storing the RS code words. However, in practice it requires much longer word-lengths for storing the decision metrics than the RS code word. Thus, it is only a small disadvantage against the proposed schemes.

Table 2: Memory requirements of the current scheme and proposed schemes.

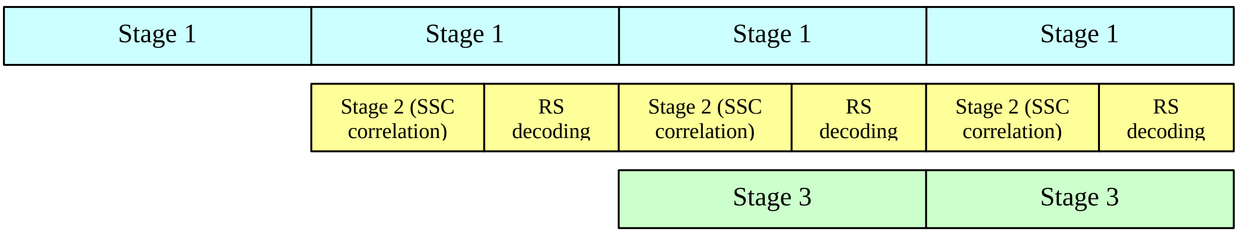
scheme	M	L	Stage 1	Stage 2	Stage 3
current scheme	32	16	2560	273+256	18
1 (preferred scheme)	256	2	2560	273+4096	4
2	128	4	2560	273+2048	6
3	64	8	2560	273+1024	10

Through decoding RS code after SSC accumulations reduces the amount of memory requirement and also the stage 2 complexity when $N_2 > 1$, it results in an increase in peak processing power when the RS decoding takes place. This issue will be addressed in the next subsection.

3.4 Peak Processing Power of Stage 2

Peak processing power of stage 2 occurs during RS decoding stage, when the accumulations of SSC correlations are completed. This problem can be solved by allocating sufficient time for RS decoding. Let X slots be the extra time allocated for RS decoding. With this extra time the cell search procedure is illustrated in the figure below. Thus, if each stage succeeds at the first trial, the cell search time is ($N_1 + N_2 + N_3 + X/16$) frames.

Figure 1: Cell search procedure.



The required processing power in stages 2 and 3 in terms of the number of operations per slot is listed in Table 3.

Table 3: Number of operations per slot during stages 2 and 3.

scheme	*M*	*L*	*X*	Stage 2 SSC correlation	Stage 2 RS decoding	Stage 3
current scheme	32	16	4	819	2176	73728
1 (preferred scheme)	256	2	4	819	17408	9216
2	128	4	4	819	8704	18432
3	64	8	4	819	4352	36864

From Table 3, the preferred scheme does introduce significantly higher processing power during stage 2 when $X=4$. However, in terms of operations per slot, the processing power required by the preferred scheme (with X=4) in stage 2 is still lower than the peak power required by the current scheme during stage 3.

Another factor to be considered is the memory access rate requirement. The peak memory access rate occurs during the RS decoding step of stage 2. The memory access rate required during RS decoding is

$$R_a = M * 272 / X /(0.625e-3) = M / X * 435200 .$$

Table 4 lists the peak memory access rate requirement.

Table 4: Peak memory access requirement.

scheme	*M*	*X*	Stage 2 RS decoding
current scheme	32	4	3.5M
1 (preferred scheme)	256	4	27.9M
2	128	4	13.9M
3	64	4	7.0M

Given a memory access clock of 32 MHz, 4 slots of RS decoding time is sufficient for the preferred scheme.

4. Simulation Results

To evaluate the performance of the proposed scheme, the link simulations are performed for each individual cell search stage. Since stage 1 is the same for all schemes, only stage 2 and 3 performance are shown. The stage 3 performance is constrained on achieving 0.5% false detection rate. It can be seen from Figure 2 that in stage 2, the preferred scheme is 1 dB worse than the current scheme. This is due to an increase in the number of nearest neighbors for the RS code used. This loss can however be compensated in stage 3. As shown in Figure 3, the preferred scheme is about 1.0 dB better compared to the current scheme, due to a decrease in number of hypotheses.

Figure 2: Stage 2 performance in a flat fading channel. (velocity= 5 km/h, $N_2 = 1$**)**

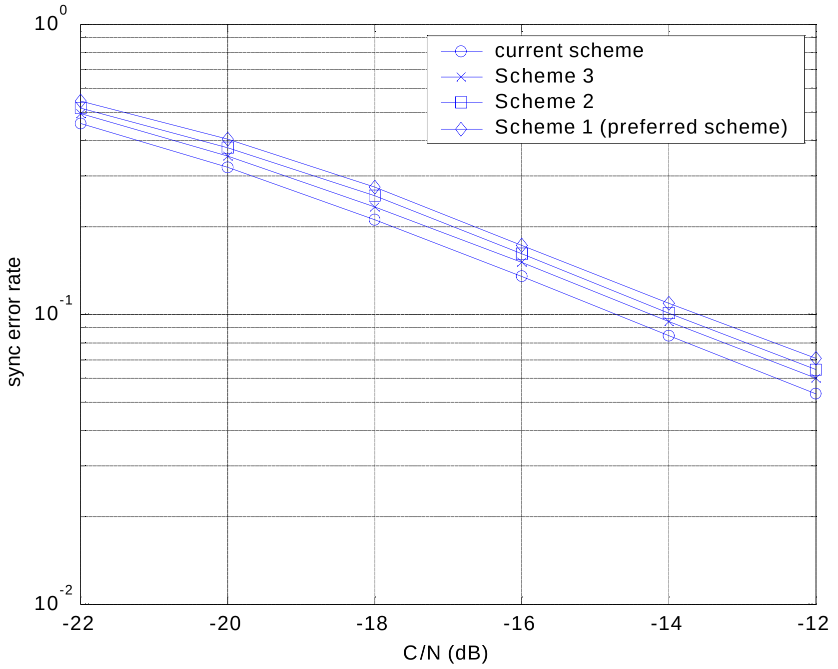
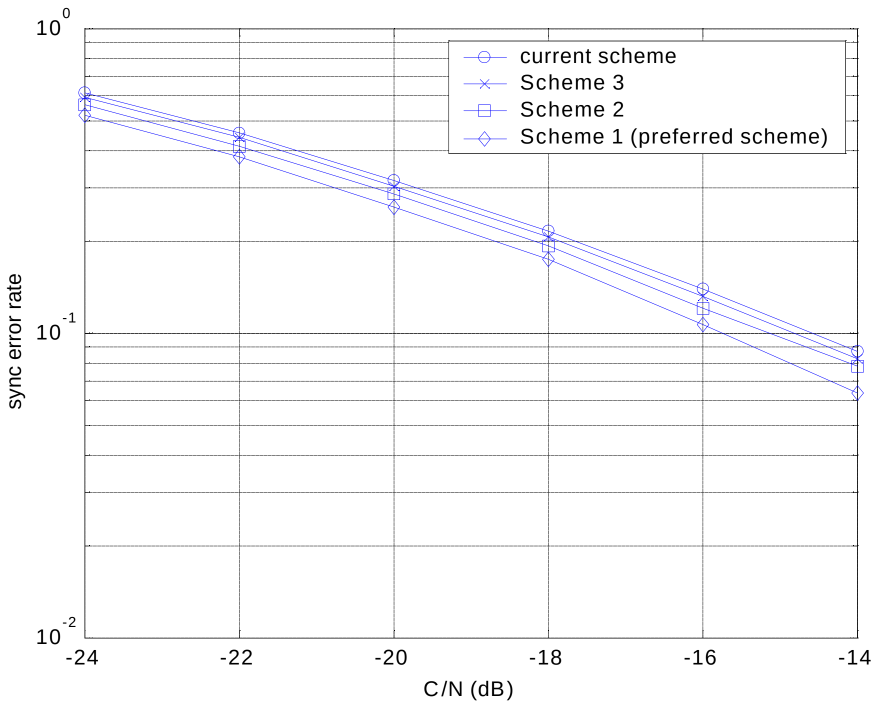


Figure 3: Stage 3 performance in a flat fading channel. (velocity= 5 km/h, $N_3 = 1$, false detection rate=0.5%**)**



5. Conclusion

In this proposal it is shown that the initial cell search can be achieved with both less complexity and less number of correlators than it is done in the current scheme in the technical specification document [2]. It is also easily understood that the proposed scheme will make it much easier for the operators when it comes to planning of the scrambling codes and the scrambling code group assignments in the purpose of keeping down the complexity in the MS. The complexity during initial cell search is reduced by 39% to 24% by the proposed schemes. The only drawback of the proposed schemes is the peak memory access rate, which occurs during the RS decoding process in stage 2. However, such a rate can be sustained by using a 32 MHz memory access clock with 4 slots RS decoding time.

References

[1]. Ericsson, "New Downlink Scrambling Code Grouping Scheme for UTRA/FDD," 3GPP TSGR1-541/99, June, 1999.

[2]. 3GPP RAN TS 25.213 V2.1.0 (1999-04). Spreading and modulation (FDD)

[3]. 3GPP RAN TS 25.214 V1.1.0 (1999-06). UTRA FDD Physical layer procedures

[4]. Christer Östberg, Yi-Pin Eric Wang and Fredrik Janecke, "Performance and Complexity of Techniques for Achieving Fast Sector Identification in an Asynchronous CDMA System," Proceedings of Personal Multimedia Communications Conference, Yukosuka, November, 1998.

[5]. Siemens and Texas Instruments, "Generalized Hierarchical Golay Sequence for PSC with Low Complexity Correlation Using Pruned Efficient Golay Correlators," 3GPP TSGR1-554/99, June, 1999.

6. Text Proposal in [2] (Preferred Scheme)

5.2.3 Synchronisation codes

5.2.3.1 Code Generation

The Primary and Secondary code words, C_p and $\{C_1,\ldots,C_{16}\}$ are constructed as the position wise addition modulo 2 of a Hadamard sequence and a fixed so called hierarchical sequence. The Primary SCH is furthermore chosen to have good aperiodic auto correlation properties.

The hierarchical sequence y is constructed from two constituent sequences x_1 and x_2 of length n_1 and n_2 respectively using the following formula:

$y(i) = x_2(i \bmod n_2) + x_1(i \operatorname{div} n_2) \text{ modulo } 2,\ i = 0 \ldots (n_1 * n_2) - 1$

The constituent sequences x_1 and x_2 are chosen to be the following length 16 (i.e. $n_1 = n_2 = 16$) sequences:

x1 = < 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1 >

and

x_2 = < 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0 >

The Hadamard sequences are obtained as the rows in a matrix H_8 constructed recursively by:

$$H_0 = (0)$$

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & \overline{H_{k-1}} \end{pmatrix}, \quad k \geq 1$$

The rows are numbered from the top starting with row *0* (the all zeros sequence).

The Hadamard sequence *h* depends on the chosen code number *n* and is denoted h_n in the sequel.

This code word is chosen from every 8th row of the matrix H_8. Therefore, there are 32 possible code words out of which 18 are used.

Furthermore, let $h_n(i)$ and $y(i)$ denote the *i:*th symbol of the sequence h_n and *y*, respectively.

Then h_n is equal to the row of H_8 numbered by the bit reverse of the 8 bit binary representation of *n*.

The definition of the *n*:th SCH code word follows (the left most index correspond to the chip transmitted first in each slot):

$C_{SCH,n} = < h_n(0)+y(0), h_n(1)+y(1), h_n(2)+y(2), \ldots, h_n(255)+y(255) >$,

All sums of symbols are taken modulo 2.

These binary code words are converted to real valued sequences by the transformation '0' -> '+1', '1' -> '-1'.

The Primary SCH and Secondary SCH code words are defined in terms of $C_{SCH,n}$ and the definition of C_p and $\{C_1,\ldots,C_{16}\}$ now follows as:

$C_p = C_{SCH,0}$
and

$C_i = C_{SCH,i}$, i=1,…,16
The definitions of C_p and $\{C_1,\ldots,C_{16}\}$ are such that a 32 point fast Hadamard transform can be utilised for detection.

5.2.3.2 Code Allocation

The 256 sequences are constructed such that their cyclic-shifts are unique, i.e., a non-zero cyclic shift less than 15 of any of the 256 sequences is not equivalent to some cyclic shift of any other of the 256 sequences. Also, a non-zero cyclic shift less than 15 of any of the sequences is not equivalent to itself with any other cyclic shift less than 16. The following sequences are used to encode the 256 different code groups each containing 1 scrambling code (note that “i” indicates the i’th Secondary Short code of the 16 codes). Note that a Secondary Short code can be different from one time slot to another and that the sequence pattern can be different from one cell to another, depending on Scrambling Code Group of Scrambling Code the cell uses

Table 9 Spreading Code allocation for Secondary SCH Code, the index “i” of the code C_i

scrambling	slot number														
Code Group	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
Group 1	1	1	2	8	9	10	15	8	10	16	2	7	15	7	16
Group 2	1	1	3	15	4	2	16	15	2	14	3	13	16	13	14
Group 3	1	1	4	10	12	9	2	10	9	3	4	11	2	11	3
Group 4	1	1	5	16	7	3	14	16	3	10	5	12	14	12	10
Group 5	1	1	6	9	15	12	4	9	12	7	6	14	4	14	7
Group 6	1	1	7	2	6	4	3	2	4	5	7	8	3	8	5
Group 7	1	1	8	7	14	11	13	7	11	12	8	2	13	2	12
Group 8	1	1	9	14	13	5	10	14	5	2	9	6	10	6	2
Group 9	1	1	10	11	5	14	8	11	14	15	10	4	8	4	15
Group 10	1	1	11	4	16	6	7	4	6	13	11	10	7	10	13
Group 11	1	1	12	5	8	13	9	5	13	4	12	16	9	16	4
Group 12	1	1	13	3	11	7	5	3	7	9	13	15	5	15	9
Group 13	1	1	14	6	3	16	11	6	16	8	14	9	11	9	8
Group 14	1	1	15	13	10	8	12	13	8	6	15	3	12	3	6
Group 15	1	1	16	12	2	15	6	12	15	11	16	5	6	5	11
Group 16	1	2	1	15	5	5	12	16	6	11	2	16	11	15	12
Group 17	1	2	2	10	13	14	6	9	13	6	1	10	5	9	5
Group 18	1	2	3	1	8	6	5	2	5	8	4	4	6	3	7
Group 19	1	2	4	8	16	13	11	7	14	9	3	6	12	5	10
Group 20	1	2	5	2	3	7	7	1	8	4	6	5	8	6	3
Group 21	1	2	6	7	11	16	9	8	15	13	5	3	10	4	14
Group 22	1	2	7	16	2	8	10	15	7	15	8	9	9	10	16
Group 23	1	2	9	4	9	1	3	3	2	12	10	11	4	12	11
Group 24	1	2	10	5	1	10	13	6	9	5	9	13	14	14	6
Group 25	1	2	11	14	12	2	14	13	1	7	12	7	13	8	8
Group 26	1	2	12	11	4	9	4	12	10	10	11	1	3	2	9
Group 27	1	2	13	13	15	3	16	14	4	3	14	2	15	1	4
Group 28	1	2	14	12	7	12	2	11	11	14	13	8	1	7	13
Group 29	1	2	15	3	14	4	1	4	3	16	16	14	2	13	15
Group 30	1	2	16	6	6	11	15	5	12	1	15	12	16	11	2
Group 31	1	3	1	16	9	9	6	14	11	8	3	14	8	16	6
Group 32	1	3	4	7	4	1	5	5	3	6	2	8	7	6	8
Group 33	1	3	5	1	15	11	9	3	9	15	7	7	11	5	13
Group 34	1	3	6	8	7	4	7	6	2	2	8	1	5	3	4

Group 35	1	3	7	15	14	12	8	13	10	4	5	11	6	9	2
Group 36	1	3	8	10	6	3	10	12	1	13	6	13	12	15	15
Group 37	1	3	9	3	5	13	13	1	15	7	11	9	15	11	5
Group 38	1	3	10	6	13	6	3	8	8	10	12	15	1	13	12
Group 39	1	3	11	13	8	14	4	15	16	12	9	5	2	7	10
Group 40	1	3	12	12	16	5	14	10	7	5	10	3	16	1	7
Group 41	1	3	13	14	3	15	2	16	13	16	15	4	4	2	14
Group 42	1	3	14	11	11	8	16	9	6	1	16	6	14	8	3
Group 43	1	3	16	5	10	7	1	7	5	14	14	10	3	12	16
Group 44	1	4	2	7	5	6	1	6	7	3	3	5	4	8	2
Group 45	1	4	4	9	8	5	16	12	8	16	1	9	13	12	13
Group 46	1	4	5	15	11	15	4	14	14	5	8	10	1	11	8
Group 47	1	4	6	10	3	8	14	11	5	12	7	16	15	13	9
Group 48	1	4	7	1	10	16	13	4	13	10	6	6	16	7	11
Group 49	1	4	8	8	2	7	3	5	6	7	5	4	2	1	6
Group 50	1	4	9	13	1	9	8	16	12	13	12	8	5	5	16
Group 51	1	4	11	3	4	10	9	2	11	2	10	12	12	9	3
Group 52	1	4	12	6	12	1	7	7	4	15	9	14	6	15	14
Group 53	1	4	13	4	7	11	11	1	10	6	16	13	10	16	7
Group 54	1	4	14	5	15	4	5	8	1	11	15	11	8	10	10
Group 55	1	4	15	14	6	12	6	15	9	9	14	1	7	4	12
Group 56	1	4	16	11	14	3	12	10	2	8	13	7	9	6	5
Group 57	1	5	1	14	4	4	11	10	8	15	5	10	15	14	11
Group 58	1	5	2	11	12	11	5	15	15	2	6	16	1	12	6
Group 59	1	5	4	5	9	12	12	1	16	13	8	4	16	8	9
Group 60	1	5	6	6	14	9	10	2	13	9	2	5	14	1	13
Group 61	1	5	7	13	7	1	9	9	5	11	3	15	13	11	15
Group 62	1	5	8	12	15	10	7	16	14	6	4	9	3	13	2
Group 63	1	5	9	1	16	8	4	5	4	16	13	13	8	9	12
Group 64	1	5	10	8	8	15	14	4	11	1	14	11	10	15	5
Group 65	1	5	11	15	13	7	13	11	3	3	15	1	9	5	7
Group 66	1	5	12	10	5	16	3	14	12	14	16	7	7	3	10
Group 67	1	5	13	16	10	6	15	12	2	7	9	8	11	4	3
Group 68	1	5	14	9	2	13	1	13	9	10	10	2	5	6	14
Group 69	1	5	15	2	11	5	2	6	1	12	11	12	6	16	16
Group 70	1	6	2	5	16	15	16	2	12	12	5	1	11	6	15
Group 71	1	6	3	14	5	7	15	9	4	10	8	11	12	16	13
Group 72	1	6	4	11	13	16	1	16	11	7	7	13	6	10	4
Group 73	1	6	5	13	2	6	13	10	1	14	2	14	10	9	9
Group 74	1	6	6	12	10	13	3	15	10	3	1	12	8	15	8
Group 75	1	6	8	6	11	14	14	1	9	16	3	8	9	3	11
Group 76	1	6	10	10	4	11	7	13	16	11	13	6	4	1	16
Group 77	1	6	11	1	9	3	8	6	8	9	16	16	3	11	14
Group 78	1	6	12	8	1	12	10	3	15	8	15	10	13	13	3
Group 79	1	6	13	2	14	2	6	5	5	13	10	9	1	14	10
Group 80	1	6	14	7	6	9	12	4	14	4	9	15	15	12	7
Group 81	1	6	15	16	15	1	11	11	6	2	12	5	16	2	5
Group 82	1	6	16	9	7	10	5	14	13	15	11	3	2	8	12
Group 83	1	7	3	13	9	11	1	11	13	5	5	9	7	15	3
Group 84	1	7	6	11	6	1	13	13	7	16	4	10	11	16	10

Group 85	1	7	8	5	7	2	4	3	8	3	2	6	6	4	5
Group 86	1	7	9	16	8	16	7	10	10	9	15	2	1	8	15
Group 87	1	7	10	9	16	7	9	15	1	8	16	8	15	2	2
Group 88	1	7	11	2	5	15	10	8	9	6	13	14	16	12	4
Group 89	1	7	14	8	10	5	6	2	3	15	12	13	4	11	9
Group 90	1	7	15	15	3	13	5	9	11	13	9	7	3	1	11
Group 91	1	7	16	10	11	6	11	16	4	4	10	1	13	7	6
Group 92	1	8	1	13	16	16	5	12	9	4	8	12	4	13	5
Group 93	1	8	2	12	8	7	11	13	2	13	7	14	14	11	12
Group 94	1	8	3	3	13	15	12	6	10	15	6	8	13	1	10
Group 95	1	8	5	4	10	14	10	5	11	11	4	1	15	8	14
Group 96	1	8	8	11	3	6	9	14	3	9	1	11	16	14	16
Group 97	1	8	9	2	4	12	14	7	13	3	16	15	11	10	6
Group 98	1	8	10	7	12	3	4	2	6	14	15	9	5	16	11
Group 99	1	8	11	16	1	11	3	9	14	16	14	3	6	6	9
Group 100	1	8	12	9	9	4	13	16	5	1	13	5	12	4	8
Group 101	1	8	13	15	6	10	1	10	15	12	12	6	8	3	13
Group 102	1	8	14	10	14	1	15	15	8	5	11	4	10	5	4
Group 103	1	9	1	10	7	7	8	2	15	16	9	2	16	10	8
Group 104	1	9	2	15	15	16	10	7	8	1	10	8	2	16	9
Group 105	1	9	4	1	14	15	7	9	7	14	12	12	15	4	6
Group 106	1	9	6	2	9	14	5	10	6	10	14	13	13	5	2
Group 107	1	9	7	9	4	6	6	1	14	12	15	7	14	15	4
Group 108	1	9	10	4	3	12	1	12	4	2	2	3	9	11	10
Group 109	1	9	11	11	10	4	2	3	12	4	3	9	10	1	12
Group 110	1	9	12	14	2	11	16	6	3	13	4	15	8	7	5
Group 111	1	9	15	6	16	2	13	14	10	11	7	4	5	12	3
Group 112	1	10	2	1	11	12	3	10	3	11	9	9	12	2	4
Group 113	1	10	3	10	2	4	4	1	11	9	12	3	11	12	2
Group 114	1	10	4	15	10	11	14	8	4	8	11	5	5	14	15
Group 115	1	10	9	11	15	7	6	4	16	5	2	12	13	3	14
Group 116	1	10	10	14	7	16	12	5	7	12	1	14	3	5	3
Group 117	1	10	12	4	6	15	5	11	8	7	3	2	14	9	16
Group 118	1	10	14	3	1	14	7	12	5	3	5	7	16	16	12
Group 119	1	11	2	2	7	8	13	12	14	8	12	11	7	1	14
Group 120	1	11	5	10	9	13	16	4	7	2	15	8	6	14	12
Group 121	1	11	7	8	12	14	1	14	8	13	13	12	11	2	7
Group 122	1	11	10	13	11	4	6	7	10	7	4	16	16	6	13
Group 123	1	11	14	4	13	2	9	10	12	16	8	5	3	15	6
Group 124	1	12	2	16	3	4	8	5	9	14	11	6	13	15	7
Group 125	1	12	3	7	10	12	7	14	1	16	10	16	14	5	5
Group 126	1	12	7	10	16	10	12	3	3	7	14	5	1	16	14
Group 127	1	12	9	6	7	15	1	15	6	4	4	7	12	14	9
Group 128	1	12	12	13	14	7	2	8	14	2	1	13	11	8	11
Group 129	1	12	13	11	1	13	14	2	8	11	8	14	7	7	2
Group 130	1	12	14	14	9	6	4	7	15	6	7	12	9	1	15
Group 131	1	12	15	5	4	14	3	16	7	8	6	2	10	11	13
Group 132	1	12	16	4	12	5	13	9	16	9	5	8	8	13	4
Group 133	1	13	2	4	14	13	4	16	1	15	14	15	16	3	3
Group 134	1	13	4	14	15	14	13	2	2	4	16	3	1	15	16

Group 135	1	13	8	3	9	16	2	15	4	11	12	10	14	6	7
Group 136	1	13	10	15	2	9	11	3	5	16	6	12	7	8	4
Group 137	1	13	15	9	13	3	7	5	15	5	3	11	11	7	9
Group 138	1	14	5	6	8	4	12	9	15	3	10	13	7	2	16
Group 139	1	14	9	8	14	6	16	11	9	11	6	3	3	16	8
Group 140	1	14	13	9	12	8	3	6	11	4	2	10	16	5	15
Group 141	1	14	14	16	4	15	13	3	4	13	1	16	2	3	2
Group 142	1	14	16	2	1	16	4	13	3	2	3	4	15	15	13
Group 143	1	15	2	13	6	5	7	3	11	10	16	4	9	14	8
Group 144	1	15	3	6	15	13	8	12	3	12	13	10	10	8	6
Group 145	1	15	4	3	7	6	10	13	12	5	14	16	8	2	11
Group 146	1	15	9	7	2	10	2	9	8	8	7	1	16	15	10
Group 147	1	15	10	2	10	1	16	16	15	9	8	7	2	9	7
Group 148	1	16	3	12	11	9	13	5	8	2	14	7	4	10	15
Group 149	1	16	5	11	16	12	15	6	5	6	12	2	2	15	11
Group 150	1	16	7	5	13	11	2	12	6	9	10	14	15	3	8
Group 151	2	2	5	10	16	11	3	10	11	8	5	13	3	13	8
Group 152	2	2	6	15	8	4	13	15	4	9	6	11	13	11	9
Group 153	2	2	9	12	6	13	7	12	13	16	9	3	7	3	16
Group 154	2	2	11	6	7	14	10	6	14	3	11	15	10	15	3
Group 155	2	2	12	3	15	5	8	3	5	14	12	9	8	9	14
Group 156	2	2	13	5	4	15	12	5	15	7	13	10	12	10	7
Group 157	2	2	14	4	12	8	6	4	8	10	14	16	6	16	10
Group 158	2	2	16	14	9	7	11	14	7	5	16	4	11	4	5
Group 159	2	3	3	10	7	6	15	11	7	15	2	10	14	11	14
Group 160	2	3	5	9	4	7	13	12	6	11	8	15	16	14	10
Group 161	2	3	6	16	12	16	3	13	13	6	7	9	2	12	7
Group 162	2	3	8	2	9	15	14	3	14	9	5	5	15	8	12
Group 163	2	3	10	14	2	10	7	15	11	14	11	7	6	6	15
Group 164	2	3	11	5	11	2	8	8	3	16	10	13	5	16	13
Group 165	2	3	13	6	16	3	6	7	2	12	16	12	7	9	9
Group 166	2	3	14	3	8	12	12	2	9	5	15	14	9	15	8
Group 167	2	3	16	13	5	11	5	16	10	10	13	2	8	3	11
Group 168	2	4	2	15	10	10	5	13	12	7	4	13	7	15	5
Group 169	2	4	6	2	16	12	10	4	10	16	8	8	12	6	14
Group 170	2	4	7	9	5	4	9	11	2	14	5	14	11	16	16
Group 171	2	4	9	5	14	5	4	7	7	9	11	16	2	14	11
Group 172	2	4	10	4	6	14	14	2	16	8	12	10	16	12	6
Group 173	2	4	11	11	15	6	13	9	8	6	9	4	15	2	8
Group 174	2	4	13	12	12	7	15	10	5	2	15	5	13	7	4
Group 175	2	4	15	6	9	8	2	8	6	13	13	9	4	11	15
Group 176	2	5	3	12	14	15	2	15	12	8	8	14	5	9	3
Group 177	2	5	4	13	6	8	16	10	3	9	7	12	11	15	14
Group 178	2	5	5	11	9	14	4	16	9	4	2	11	7	16	7
Group 179	2	5	7	5	12	13	13	2	10	15	4	7	10	4	12
Group 180	2	5	9	9	3	12	8	14	15	12	14	5	3	2	15
Group 181	2	5	11	7	2	11	9	4	16	7	16	9	14	14	4
Group 182	2	5	12	2	10	4	7	5	7	10	15	15	4	12	13
Group 183	2	5	13	8	5	10	11	3	13	3	10	16	16	11	8
Group 184	2	6	2	13	3	3	12	9	7	16	6	9	16	13	12

Group 185	2	6	3	6	10	11	11	2	15	14	7	3	15	7	10
Group 186	2	6	8	14	8	2	10	10	6	12	4	16	14	12	16
Group 187	2	6	9	7	7	16	13	3	12	2	13	12	9	16	6
Group 188	2	6	10	2	15	7	3	6	3	15	14	14	7	10	11
Group 189	2	6	11	9	6	15	4	13	11	13	15	8	8	4	9
Group 190	2	6	12	16	14	8	14	12	4	4	16	2	10	6	8
Group 191	2	7	2	14	15	15	6	11	10	3	7	11	3	14	6
Group 192	2	7	4	4	14	16	11	5	9	16	5	7	14	2	9
Group 193	2	7	6	3	9	13	9	6	12	12	3	2	16	7	13
Group 194	2	7	7	12	4	5	10	13	4	10	2	12	15	13	15
Group 195	2	7	11	10	10	3	14	15	6	2	14	6	11	3	7
Group 196	2	7	12	15	2	12	4	10	13	15	13	4	5	5	10
Group 197	2	7	13	9	13	2	16	16	7	6	12	3	9	6	3
Group 198	2	7	14	16	5	9	2	9	16	11	11	5	7	4	14
Group 199	2	8	4	14	10	12	2	12	14	6	6	10	8	16	4
Group 200	2	8	5	12	5	2	14	14	8	15	3	9	12	15	9
Group 201	2	8	15	9	12	5	12	15	3	3	9	2	14	8	5
Group 202	2	8	16	16	4	14	6	10	12	14	10	8	4	2	12
Group 203	2	9	3	16	9	12	13	7	3	7	12	6	6	13	16
Group 204	2	9	9	13	8	15	11	6	8	11	2	13	4	6	4
Group 205	2	9	13	4	2	13	8	11	6	4	6	8	15	15	11
Group 206	2	10	3	2	13	16	8	10	8	13	11	11	16	3	5
Group 207	2	10	8	10	3	5	5	2	13	11	16	8	13	16	3
Group 208	2	11	4	8	9	11	8	13	2	15	9	15	13	6	6
Group 209	2	11	8	9	15	9	11	4	4	8	13	6	2	15	13
Group 210	2	11	10	5	8	16	2	16	5	3	3	8	11	13	10
Group 211	2	11	13	13	10	5	3	8	16	5	8	11	10	2	16
Group 212	2	11	15	3	11	6	14	10	15	10	6	7	7	14	3
Group 213	2	12	9	14	12	3	5	8	9	8	3	15	15	5	14
Group 214	2	13	10	7	13	5	15	12	10	12	5	4	4	15	7
Group 215	2	14	16	10	14	4	8	6	16	6	4	12	12	8	10
Group 216	2	16	4	5	16	14	7	11	4	11	14	9	9	7	5
Group 217	3	3	4	6	11	12	13	6	12	14	4	5	13	5	14
Group 218	3	3	6	5	16	9	15	5	9	10	6	4	15	4	10
Group 219	3	3	11	16	15	7	12	16	7	4	11	8	12	8	4
Group 220	3	3	14	10	4	13	8	10	13	9	14	7	8	7	9
Group 221	3	4	3	13	7	7	10	14	8	9	4	14	9	13	10
Group 222	3	4	4	12	15	16	8	11	15	8	3	12	7	11	7
Group 223	3	4	5	14	4	6	12	13	5	13	6	11	11	12	14
Group 224	3	4	9	16	10	4	16	15	3	5	10	5	15	6	6
Group 225	3	4	12	7	3	12	15	8	11	7	11	15	16	16	8
Group 226	3	4	14	8	8	9	13	7	10	3	13	10	14	9	4
Group 227	3	4	16	10	5	10	4	9	9	16	15	6	3	5	15
Group 228	3	5	11	14	6	14	5	12	12	11	13	4	3	6	13
Group 229	3	5	12	11	14	5	11	13	3	6	14	6	13	4	4
Group 230	3	6	4	10	6	5	9	15	4	15	5	16	16	9	10
Group 231	3	7	4	9	10	9	7	13	13	4	8	14	3	10	8
Group 232	3	7	8	8	16	11	12	4	15	11	4	7	16	3	15
Group 233	3	7	13	4	9	7	4	8	3	10	9	10	8	14	14
Group 234	3	7	16	11	4	15	3	15	11	12	12	4	7	8	16

Group 235	3	8	4	7	14	13	14	4	10	10	7	3	9	8	13
Group 236	3	8	7	15	4	8	15	12	3	16	4	16	12	11	11
Group 237	3	8	13	14	13	3	9	9	8	4	10	7	14	4	7
Group 238	3	8	15	4	16	4	8	7	7	15	12	11	3	16	12
Group 239	3	9	4	4	5	6	15	10	16	6	10	9	5	3	16
Group 240	3	9	5	6	10	16	3	16	6	15	15	10	9	4	5
Group 241	3	10	10	15	16	5	4	6	16	4	3	15	9	6	9
Group 242	3	10	15	9	3	15	16	4	6	9	6	16	5	5	4
Group 243	3	11	3	12	5	5	6	4	13	14	11	4	14	12	6
Group 244	3	11	4	13	13	14	12	5	6	3	12	6	4	14	11
Group 245	3	11	8	4	11	16	7	12	8	12	16	15	15	7	4
Group 246	3	13	11	5	4	12	4	11	6	6	5	3	14	13	12
Group 247	3	13	12	4	12	3	14	14	13	11	6	5	4	11	5
Group 248	3	14	7	9	14	10	13	8	7	8	10	4	4	13	9
Group 249	5	5	7	11	8	6	12	11	6	10	7	9	12	9	10
Group 250	5	5	8	14	16	13	6	14	13	7	8	15	6	15	7
Group 251	5	6	9	9	11	7	12	10	8	7	10	6	11	5	8
Group 252	5	6	11	7	10	8	5	8	7	12	12	10	6	9	11
Group 253	5	6	13	8	13	5	7	7	6	16	14	15	8	16	15
Group 254	5	6	16	15	8	13	8	16	14	14	15	5	7	6	13
Group 255	5	7	9	10	7	11	6	12	9	12	11	8	8	6	10
Group 256	5	8	15	7	8	14	13	6	15	6	14	16	16	13	7
Sync BTS	5	9	6	8	10	9	8	12	5	11	10	11	12	7	7