4G/LTE
[+]Quick Reference
Quick Picture
Basic Procedure
Troubleshoot Tips
Frame Structure
Downlink
Uplink
Reference Signal
Low Layer Procedures
DCI
HARQ
PHY Processing
MIMO
MIMO/CA(Big Pict.)
Precoding
Fading
Power Control
PUCCH
Protocols
RACH
SIB
PAGING
MAC
RLC
PDCP
Full Stack
[+] Basic Call
Idle Mode
Multi Cell
UE Capability
Conformance
LTE RF/PHY
RF Measurement
LTE TDD
Overview
LTE IoT
LTE-M1
NB-IoT

4G/LTE - RACH

Follow
6,092

LTE RACH In a Nutshell

- What is it for ? It is for establishing Uplink Synchronization
- Two types of RACH Process : Contention based and Non Contention Based
- Possible Subcarrier Spacing of RACH Preamble : 1.25 Khz only
- Preamble format and Sequence Length : Format 0,1,2,3,4
- What are main differences among the Preamble Formats ? : Length of Preamble
- Why are there different Preamble types ? The biggest reasons are to cover various range of distances between UE and Cell and to enhance reliable reception of the preamble with the optimal use of network resources.
- RRC Parameter to determine Preamble Type, Sequence Length, PRACH Transmission Time : prach-ConfigIndex

LTE RACH in Details

RACH stands for **R**andom **A**ccess **Ch**annel. This is the first message from UE to eNB when you power it on. Even though they use a little bit different name, in all cellular technology (CDMA, GSM, WCDMA, LTE) there is a specific signal that perform the same function. In CDMA, they call it 'Access Probe' as far as I know (but I don't have much knowledge on CDMA), in GSM they call it 'Channel Request', and in WCDMA / LTE they call it 'RACH'. In terms of eNB point of view, it would seem that it is getting this initial UE signal in almost random fashion (e.g, in Random timing , Random Frequency and in Random Identification) because it has no idea when a user turn on the UE (Actually it is not completely random, there are a certain range of agreement between UE and Network about the timing, frequency location and possible indentification, but in large scale it would look like working in random fashion). In terms of Radio Access Network implementation, handling RACH would be one of the most challenging job. Even in terms of protocol design, RACH design can be one of the most important / critical portions.
If you have any experience of testing UE or UE modem chipset at the very early stage of the development, you would have noticed that RACH is the most challenging point in terms of troubleshooting. Why ?
Before UE decided to send RACH signal (RACH preamble), there are many preconditions to be met as described in From Power-On to PRACH. If there are some problems in precondion stage, you have to completely rely on UE side log only, Network side log does not help anything. Even though you have access to the UE side log, depending on UE modem chipset.. some provides pretty detailed information but others does not provide much detailed information. Even if the UE log provides the detailed information, there are not many people who can properly interpret those information and find out the root cause of the problem.
You would see a lot of issues and spend many many stressful days when you are testing the device at its early stage of the implementation, but once the device / modem is mature you would even forget about the fact that there is such a step called 'RACH' because this wouldn't cause any trouble.

Anyway... in short. RACH is one of the most important steps in LTE protocol (actually in any Cellular protocol) and there are a lot of details to be covered as below.

- Why RACH ? (What is the functionality of RACH ?)
- When RACH Process occurs ?
- Two types of RACH process : Contention-based and Contention-free
- How the information is encoded into PRACH (RACH Preamble) ?
- Exactly when and Where a UE transmit RACH ?
- What is preamble format ?
- Why Multiple Preamble Format ?
- How to determined which Preamble format to use ?
- How does Network knows exactly when UE will transmit the RACH ?
- How many RA-Preambles can be used ?
- How to Generate 64 PRACH Preamble Sequences ?
- PRACH Preamble Signal Structure
- How to generate RACH Signal ?
  - PRACH Sequence in Frequency Domain
  - PRACH Sequence in Time Domain
- Exactly when and where Network transmit RACH Response
- How is the RACH Preamble Power determined ?
  - Example
- PRACH Parameters and it's Physical Meaning
  - prach-ConfigIndex
  - zeroCorrelationZoneConfig and Highspeedflag
  - prach-FreqOffset
  - rootSequenceIndex
- RACH Procedure during Initial Registration - RACH Procedure Summary
- How can we get RA RNTI ?
- An Example of Full RACH Process
- PRACH Retransmission
- RACH Process Overview In Diagrams

LG Elecs. Ex. 1038
LG Elecs. v. Pantech Corp.
IPR2023-01267 Page 1

- RACH Procedure on Initial Registration
  - Livenetwork Example of RACH process for Initial Attach
- RACH Procedure on Handover - Contention Based
- RACH Procedure on Handover - NonContention Based
- RACH Procedure on DL Data Arrival when Out-of-Sync - Non Contention Based
- RACH Procedure on DL Data Arrival when Out-of-Sync - Contention Based
- RACH Procedure on UL Data Arrival when Out-of-Sync
- RACH Procedure on RRC Connection Re-establishment when Out-of-Sync

- PRACH Optimization in Cell Planning
- PRACH RF Snapshot
- 3GPP Standard for RACH Process
- RACH in srsRAN

Why RACH ? (What is the functionality of RACH ?)

The first question poping up in your mind when you first hear about the word RACH or RACH Process would be 'Why RACH ?', 'What is the functionality/purpose of RACH process ?', "Why we need this kind of complicated (looks over-complicated) ?'.
For sure, it is not for confusing you :), RACH has very important functionality especially in LTE (and in WCDMA as well). The main purpose of RACH can be described as follows.

i) Achieve UP link synchronization between UE and eNB
ii) Obtain the resource for Message 3 (e.g, RRC Connection Request)

In most of the communication (especially digital comunication regardless of whether it is wired or wireless), the most important precondition is to establish the timing synchronization between the reciever and transmitter. So whatever communication technology you would study, you would see some kind of synchronization mechanism that is specially designed for the specific communication.

In LTE (in WCDMA as well), the synchronization in downlink (Transmitter = eNB, Reciever = UE), this synchronization is achieved by the special synchronization channel (special physical signal pattern). Refer to Time Sync Process page for the details. This downlink sync signal gets broadcasted to everybody and it is get transmitted all the time with a certain interval.
However in Uplink(Transmitter = UE, Reciever = eNB), it is not efficient (actually waste of energy and causing a lot of interference to other UEs) if UE is using this kind of broadcasting/always-on synchronization mechanism. You may easily understand this kind of problem. In case of uplink, this synchronization process should meet following criteria

i) The synchronization process should happen only when there is immediate necessity
ii) The synchronization should be dedicated to only a specific UE

All the complicated/confusing stories in this page is mostly about the process specially designed mechanism to meet these criteria.

Another purpose of RACH process is to obtain the resource for Msg3 (Message 3). RRC Connection Request is one example of Msg3 and there are several different types of Msg3 depending on situation. You would figure out this part in reading through this page and this is not very complicated to understand.

When RACH Process occurs ?

It would be helpful to understand if you think about when 'RRC Connection' happens (or when PRACH process happens if you are interested in lower layer stuffs) in WCDMA. It would also be helpful if you think about when 'Channel Request' happens in GSM UE.

My impression of LTE RACH process is like the combination of PRACH process (WCDMA) and Channel Request (GSM). It may not be 100% correct analogy.. but anyway I got this kind of impression. In LTE, RACH process happens in following situation (3GPP specification, 10.1.5 Random Access Procedure of 36.300)

i) Initial access from RRC_IDLE
ii) RRC Connection Re-establishment procedure
iii) Handover (Contention Based or Non Contetion Based)
iv) DL data arrival during RRC_CONNECTED requiring random access procedure
E.g. when UL synchronisation status is “non-synchronised”
v) UL data arrival during RRC_CONNECTED requiring random access procedure
E.g. when UL synchronisation status is "non-synchronised" or there are no PUCCH resources for SR available.
vi) For positioning purpose during RRC_CONNECTED requiring random access procedure;
E.g. when timing advance is needed for UE positioning

Two types of RACH process : Contention-based and Contention-free

When a UE transmit a PRACH Preamble, it transmits with a specific pattern and this specific pattern is called a "Signature". In each LTE cell, total 64 preamble signatures are available and UE select randomly one of these signatures.

UE select "Randomly" one of these signatures ?

Does this mean that there is some possibility that multiple UEs send PRACH with identical signatures ?

Yes.

There is such a possibility. It means the same PRACH preamble from multipe UE reaches the NW at the same time.. this kind of PRACH collision is called "Contention" and the RACH process that allows this type of "Contention" is called "Contention based" RACH Process. In this kind of contention based RACH process, Network would go through additional process at later step to resolve these contention and this process is called "Contention Resolution" step.

But there is some cases that these kind of contention is not acceptable due to some reason (e.g, timing restriction) and these contention can be prevented. Usually in this case, the Network informs each of the UE of exactly when and which preamble signature it has to use. Of course, in this case Network will allocate these preamble signature so that

it would not collide. This kind of RACH process is called "Contention Free" RACH procedure. To initiate the "Contention Free" RACH process, UE should be in Connected Mode before the RACH process as in Handover case.

Typical 'Contention Based' RACH Procedure is as follows :

i) UE --> NW : RACH Preamble (RA-RNTI, indication for L2/L3 message size)
ii) UE <-- NW : Random Access Response (Timing Advance, T_C-RNTI, UL grant for L2/L3 message)
iii) UE --> NW : L2/L3 message
iv) Message for early contention resolution

Now let's assume that a contention happened at step i). For example, two UEs sent PRACH. In this case, both of the UE will recieve the same T_C-RNTI and resource allocation at step ii). And as a result, both UE would send L2/L3 message through the same resource allocation(meaning with the same time/frequency location) to NW at step iii). What would happen when both UE transmit the exact same information on the exact same time/frequency location ? One possibility is that these two signal act as interference to each other and NW decode neither of them. In this case, none of the UE would have any response (HARQ ACK) from NW and they all think that RACH process has failed and go back to step i). The other possibility would be that NW could successfully decode the message from only one UE and failed to decode it from the other UE. In this case, the UE with the successful L2/L3 decoding on NW side will get the HARQ ACK from Network. This HARQ ACK process for step iii) message is called "contention resolution" process.

Typical 'Contention Free' RACH Procedure is as follows :

i) UE <--NW : RACH Preamble (PRACH) Assignment
ii) UE --> NW : RACH Preamble (RA-RNTI, indication for L2/L3 message size)
iii) UE <--NW : Random Access Response (Timing Advance, C-RNTI, UL grant for L2/L3 message)

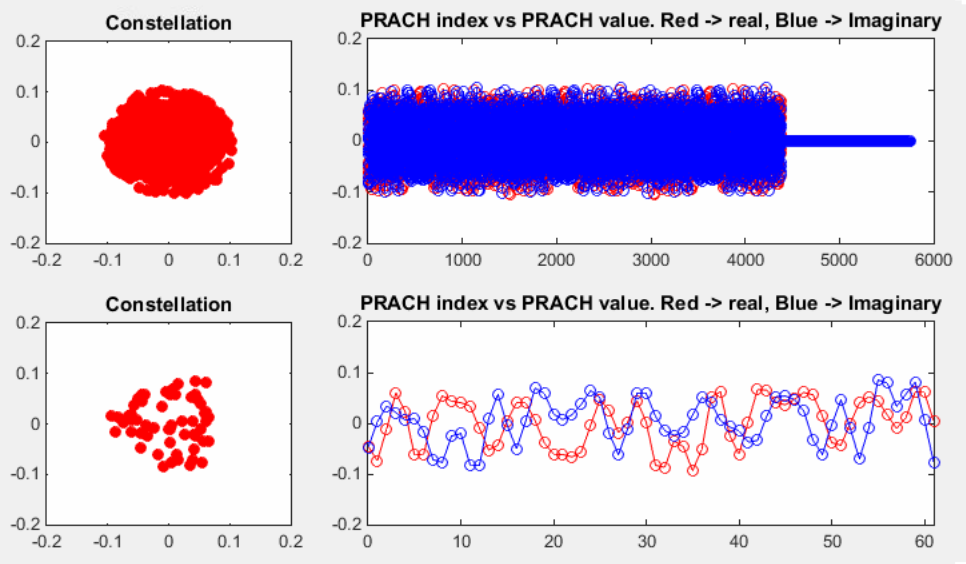
How the information is encoded into PRACH (RACH Preamble) ?

In LTE, all the information (data) after PRACH Preamble has its own binary structure meaning that they are translated into a certain data structure. However, the information in PRACH Preamble is represented by purely physical properties. The physical properties that forms the information in PRACH are as follows.

i) PRACH Preamble transmission Timing (t_id)
ii) Location of PRACH transmission in frequency domain (f_id)
iii) Sequence of the whole I/Q data of PRACH signal (one example shown below)

Following is the PRACH signal transmitted in Time Domain. (You may think this looks different from what you expected. You might have expect to see Zadoff-Chu sequence but this does not look like a Zadoff-Chu sequence. The Zadoff-Chu sequence for PRACH is in Frequency Domain, but this is the time domain sequence. The PRACH Zadoff-Chu is transformed to the time domain sequence as shown below via a transformation that is shown in How to Generate RACH signal section)

< PRACH Baseband Data Sequence >



From item i) and ii), RA_RNTI is deribed as described in How can we get RA RNTI . From item iii), Preamble Index (RAPID) can be derived.
You may not have much difficulties in understanding the derivation of RA_RNTI but it would not be that simple to understand the derivation of Preamble Index part. Most of the this page with a lot of equations and complex tables are related to this process, but I don't think I did a good job to simply/clearly describe this part. I will add another short sections in the future when I have everything cleared up in my brain. In the mean time, please refer to Matlab LTE Toolbox : PRACH page to get the intuitive (sorry it may not be that intuitive :) image in your brain and read the sections about PRACH Preamble signal generation part in this page. Of course, you would not get everything with single look. Nothing comes easy in engineering :)

Exactly when and Where a UE transmit RACH ?

To answer to this question, you need to refer to 3GPP specification TS36.211 - Table 5.7.1-2. This table would give you at which frame and subframe that UE is allowed to transmit a PRACH Preamble. As you see at this table, the prach preamble timing and prach preamble type is determined by PRACH Configuration Index. The, how PRACH Configuration Index is determined ? It is determined by SIB2 parameter prach-ConfigIndex.

< TS36.211 - Table 5.7.1-2 : PRACH Configuration Index>

PRACH Configuration Index	Preamble Format	System frame number	Subframe number	PRACH Configuration Index	Preamble Format	System frame number	Subframe number
0	0	Even	1	32	2	Even	1
1	0	Even	4	33	2	Even	4
2	0	Even	7	34	2	Even	7
3	0	Any	1	35	2	Any	1
4	0	Any	4	36	2	Any	4
5	0	Any	7	37	2	Any	7
6	0	Any	1, 6	38	2	Any	1, 6
7	0	Any	2 ,7	39	2	Any	2 ,7
8	0	Any	3, 8	40	2	Any	3, 8
9	0	Any	1, 4, 7	41	2	Any	1, 4, 7
10	0	Any	2, 5, 8	42	2	Any	2, 5, 8
11	0	Any	3, 6, 9	43	2	Any	3, 6, 9
12	0	Any	0, 2, 4, 6, 8	44	2	Any	0, 2, 4, 6, 8
13	0	Any	1, 3, 5, 7, 9	45	2	Any	1, 3, 5, 7, 9
14	0	Any	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	46	N/A	N/A	N/A
15	0	Even	9	47	2	Even	9
16	1	Even	1	48	3	Even	1
17	1	Even	4	49	3	Even	4
18	1	Even	7	50	3	Even	7
19	1	Any	1	51	3	Any	1
20	1	Any	4	52	3	Any	4
21	1	Any	7	53	3	Any	7
22	1	Any	1, 6	54	3	Any	1, 6
23	1	Any	2 ,7	55	3	Any	2 ,7
24	1	Any	3, 8	56	3	Any	3, 8
25	1	Any	1, 4, 7	57	3	Any	1, 4, 7
26	1	Any	2, 5, 8	58	3	Any	2, 5, 8
27	1	Any	3, 6, 9	59	3	Any	3, 6, 9
28	1	Any	0, 2, 4, 6, 8	60	N/A	N/A	N/A
29	1	Any	1, 3, 5, 7, 9	61	N/A	N/A	N/A
30	N/A	N/A	N/A	62	N/A	N/A	N/A
31	1	Even	9	63	3	Even	9

Did you open the specification now ? It shows exactly when a UE is supposed to send RACH depending on a parameter called "PRACH Configuration Index".

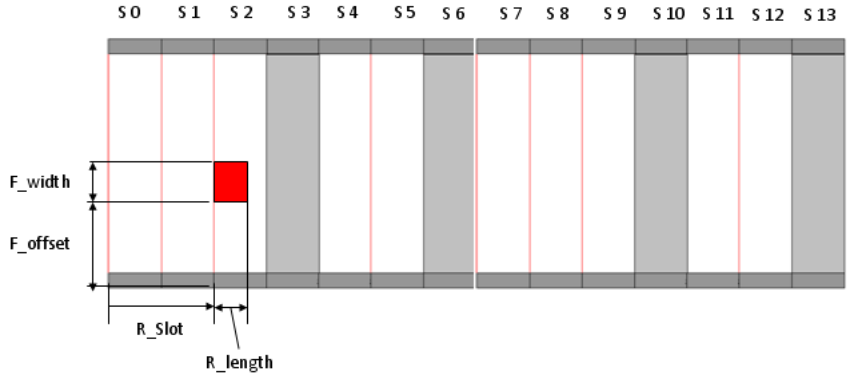
For example, if the UE is using "PRACH Configuration Idex 0", it should transmit the RACH only in EVEN number SFN(System Frame Number). Is this good enough answer ? Does this mean that this UE can transmit the RACH in any time within the specified the SFN ? The answer to this question is in "Sub Frame Number" colulmn of the table. It says "1" for "PRACH Configuration Idex 0". It means the UE is allowed to transmit RACH only at sub frame number 1 of every even SFN.

Checking your understanding of the table, I will give you one question. With which "PRACH Configuration Idex", it would be the easiest for the Network to detect the RACH from UE ? and Why ?

The answer would be 14, because UE can send the RACH in any SFN and any slots within the frame.

In a big picture, you should know all the dimmensions in the following diagram. (The Red rectangle is PRACH signal).

< PRACH Resource : Time domain and frequency domain >



The R_Slot is determined by PRACH Configuration Index and R_length is determined by Premable format. F_offset is dermined by the following equation when the preamble format is 0~3. n_RA_PRBoffset in this equation is specified by prach-FreqOffset in SIB2. (Refer to 36.211 5.7 Physical random access channel for the details)

< FDD >

$$n_{PRB}^{RA} = n_{PRB\,offset}^{RA}$$

< TDD : Preamble format 0-3 >

$$n_{PRB}^{RA} = \begin{cases} n_{PRB\ offset}^{RA} + 6\left\lfloor \frac{f_{RA}}{2} \right\rfloor, & \text{if } f_{RA} \bmod 2 = 0 \\ N_{RB}^{UL} - 6 - n_{PRB\ offset}^{RA} - 6\left\lfloor \frac{f_{RA}}{2} \right\rfloor, & \text{otherwise} \end{cases}$$

< TDD : Preamble format 4 >

$$n_{\mathrm{PRB}}^{\mathrm{RA}} = \begin{cases} 6f_{\mathrm{RA}}, & \text{if } \left((n_f \bmod 2)\times(2-N_{\mathrm{SP}}) + t_{\mathrm{RA}}^{(1)}\right) \bmod 2 = 0 \\ N_{\mathrm{RB}}^{\mathrm{UL}} - 6(f_{\mathrm{RA}}+1), & \text{otherwise} \end{cases}$$

What is preamble format ?

If you see the 36.211- table 5.7.1-1 show above, you see the column titled as "Preamble Format". What is the preamble format ? It is defined as following diagram.

You would see that the length of PRACH preamble varies depending on the preamble format. For example, the length of PRACH with preamble format 0 is (3186 + 24567) Samples. (As you know, one sample (Ts) is 1/30.72 (=0.03255) us. It is defined as 1/(15000 x 2048) seconds (=0.03255 us) in 36.211 4 Frame structure).

< PRACH Preamble Format >



Preamble format	T_{CP}	T_{SEQ}
0	$3168 \cdot T_s$	$24576 \cdot T_s$
1	$21024 \cdot T_s$	$24576 \cdot T_s$
2	$6240 \cdot T_s$	$2 \cdot 24576 \cdot T_s$
3	$21024 \cdot T_s$	$2 \cdot 24576 \cdot T_s$
4 (see Note)	$448 \cdot T_s$	$4096 \cdot T_s$
NOTE: Frame structure type 2 and special subframe configurations with UpPTS lengths $4384 \cdot T_s$ and $5120 \cdot T_s$ only assuming that the number of additional SC-FDMA symbols in UpPTS X in Table 4.2-1 is 0.		

Why Multiple Preamble Format ?

You may ask "Why we need this kind of multiple preamble format ?", especially "Why we need various PRACH format with different length in time ?".

First try to figure out what is the difference among preamble format based on the table above (Table 5.7.1-1) ? For simplicity, let's think of only format 0,1,2,3.

Let's look into T_SEQ (length of Sequence). You see format 0 and format 1 is made up of single copies of PRACH converted in time domain. Format 2 and 3 is made up of two copies of PRACH sequence concatenated. What would be the advantage that format 2,3 have over format 1,2. I think the longer T_SEQ would help decoding PRACH under noised condition because it provide longer correlation window to detect PRACH.

Now let's look at T_CP part. you would notice format 1 and 3 has much longer T_CP comparing to format 0 and 2. Longer CP would give you better tolerance in fading environment and reducing ISI even in highly fading environment.

Actually there is another important differences among each preamble format that is not explicitely shown in Table 5.7.1-1. It is guard time difference. How this guard time influence the cell size ? (Refer to the comments below or Cell Size Configuration in Random Access Procedure (I) - Preamble Format at LTE University)

I want to recommend a book titled "LTE : The UMTS From Theory to Practice" Section 19.4.2 The PRACH Structure. This is the material that describes the PRACH in the most detailed level I have ever read.

Just as a brief conclusion for cell size, we can rewrite the table as follows.

Preamble Format	T_CP (in Ts)	T_CP (in ms)	T_SEQ (in Ts)	T_SEQ (in ms)	Total Length (in ms)	Number of Subframes	Guard Time (in ms)	Cell Radius
0	3168	0.103	24576	0.800	0.903	1	0.097	~ 14 km
1	21024	0.684	24576	0.800	1.484	2	0.516	~ 75 km
2	6240	0.203	2 x 24576	1.600	1.803	2	0.197	~ 28 km
3	21024	0.684	2 x 24576	1.600	2.284	3	0.716	~ 108 km
4	448	0.015	4096	0.133	0.148			

Note 1 : T_CP (in ms) = T_CP(in Ts) x 0.03255 x 1/1000,
where 0.03225 is one Ts in us, 1/1000 is used to convert the unit from 'us' to 'ms'
Note 2 : T_SEQ (in ms) = T_SEQ(in Ts) x 0.03255 x 1/1000,
where 0.03225 is one Ts in us, 1/1000 is used to convert the unit from 'us' to 'ms'
Note 3 : Guard Time (in ms) = Number of Subframe - Total Length
Note 4 : Cell Radius is roughly the distance that the electromatic wave can travel during the guard time and devided by 2.
In case of free space(in vacumm) it is roughly is 300 (km/ms) x Guard Time (ms) / 2.

How to determined which Preamble format to use ?

How UE know which Preamble format it has to use when it generate PRACH and trnasmit ? It is determined by following table. As you see, PRACH Configuration Index determines the Preamble Format to be used. For example, if PRACH Configuration Index is 10 as shown in the following example, the preamble format 0 is used.
The you may ask 'Who determines PRACH Configuration index ?'. The answer is 'eNB determines it via prach-Configindex IE in SIB2'. Refer to prach-ConfigIndex section for the details.

< TS36.211 - Table 5.7.1-2 : Interpretation of PRACH Config >

PRACH Configuration Index	Preamble Format	System frame number	Subframe number	PRACH Configuration Index	Preamble Format	System frame number	Subframe number
0	0	Even	1	32	2	Even	1
1	0	Even	4	33	2	Even	4
2	0	Even	7	34	2	Even	7
3	0	Any	1	35	2	Any	1
4	0	Any	4	36	2	Any	4
5	0	Any	7	37	2	Any	7
6	0	Any	1, 6	38	2	Any	1, 6
7	0	Any	2 ,7	39	2	Any	2 ,7
8	0	Any	3, 8	40	2	Any	3, 8
9	0	Any	1, 4, 7	41	2	Any	1, 4, 7
10	0	Any	2, 5, 8	42	2	Any	2, 5, 8
11	0	Any	3, 6, 9	43	2	Any	3, 6, 9
12	0	Any	0, 2, 4, 6, 8	44	2	Any	0, 2, 4, 6, 8
13	0	Any	1, 3, 5, 7, 9	45	2	Any	1, 3, 5, 7, 9
14	0	Any	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	46	N/A	N/A	N/A
15	0	Even	9	47	2	Even	9
16	1	Even	1	48	3	Even	1
17	1	Even	4	49	3	Even	4
18	1	Even	7	50	3	Even	7
19	1	Any	1	51	3	Any	1
20	1	Any	4	52	3	Any	4
21	1	Any	7	53	3	Any	7
22	1	Any	1, 6	54	3	Any	1, 6
23	1	Any	2 ,7	55	3	Any	2 ,7
24	1	Any	3, 8	56	3	Any	3, 8
25	1	Any	1, 4, 7	57	3	Any	1, 4, 7
26	1	Any	2, 5, 8	58	3	Any	2, 5, 8
27	1	Any	3, 6, 9	59	3	Any	3, 6, 9
28	1	Any	0, 2, 4, 6, 8	60	N/A	N/A	N/A
29	1	Any	1, 3, 5, 7, 9	61	N/A	N/A	N/A
30	N/A	N/A	N/A	62	N/A	N/A	N/A
31	1	Even	9	63	3	Even	9

How does Network knows exactly when UE will transmit the RACH ?

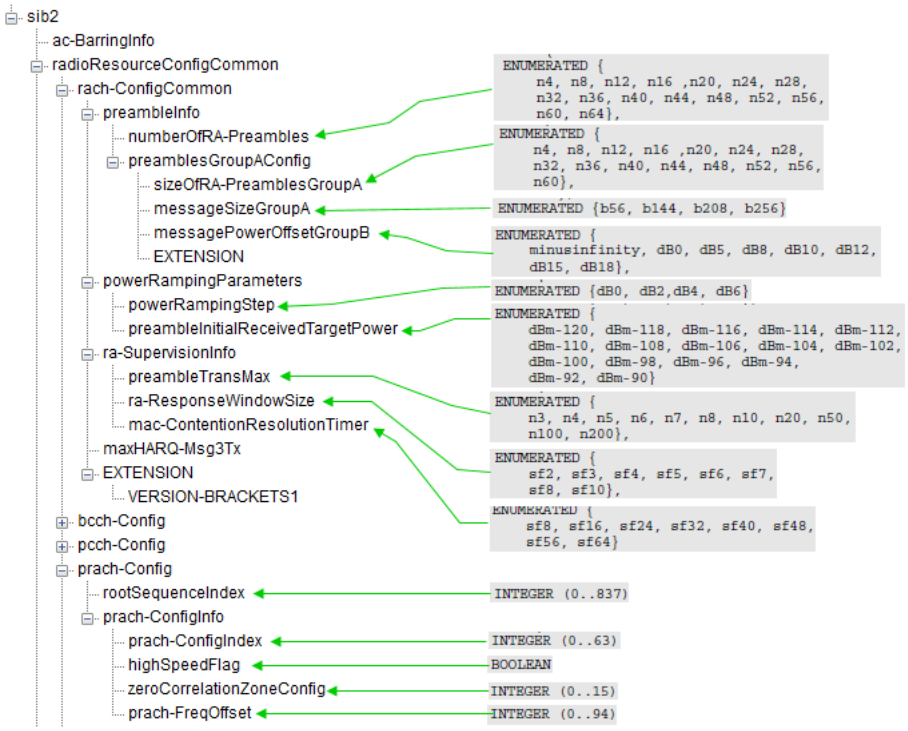
It is simple. Network knows when UE will send the RACH even before UE sends it because Network tells UE when the UE is supposed to transmit the RACH. (If UE fails to decode properly the network information about the RACH, Network will fail to detect it even though UE sends RACH).

Following section will describe network informaton on RACH.

Which RRC Message contains RACH Configuration ?

It is in SIB2 and you can find the details in 3GPP 36.331.

< RRC Parameters for RACH Operation >



How many RA-Preambles can be used ?

Theoretically 64 PRACH preambles are available in total, but the number of the preambles available in a specific condition (e.g, in a certain cell, whether it is for attach or for handover etc) are determined by a couple of SIB2 parameters as shown below.

```
sib2
   radioResourceConfigCommon
      rach-ConfigCommon
         preambleInfo
            numberOfRA-Preambles: n52 (12)
            preamblesGroupAConfig
               sizeOfRA-PreamblesGroupA: n48 (11)
               messageSizeGroupA: b56 (0)
               messagePowerOffsetGroupB: dB5 (2)
         powerRampingParameters
            powerRampingStep: dB2 (1)
            preambleInitialReceivedTargetPower: dBm-104 (8)
         ra-SupervisionInfo
            preambleTransMax: n6 (3)
            ra-ResponseWindowSize: sf10 (7)
            mac-ContentionResolutionTimer: sf48 (5)
         maxHARQ-Msg3Tx: 4
      ...
      prach-Config
         rootSequenceIndex: 22
         prach-ConfigInfo
            prach-ConfigIndex: 3
            ..0. .... highSpeedFlag: False
            zeroCorrelationZoneConfig: 5
            prach-FreqOffset: 4
      ...

      additionalSpectrumEmission: 1
   timeAlignmentTimerCommon: infinity (7)
```

There are two groups of RA-Preambles, Group A and Group B. Group A always exists and Group B exists only with the specific configuration in SIB 2 parameter. The determination of Group A and Group B is described in 36.321 5.1.1 Random Access Procedure initialization as follows.

If sizeOfRA-PreamblesGroupA is equal to numberOfRA-Preambles then there is no Random Access Preambles group B. The preambles in Random Access Preamble group A are the preambles (0 to sizeOfRAPreamblesGroupA – 1) and, if it exists, the preambles in Random Access Preamble group B are (the preambles sizeOfRA-PreamblesGroupA to numberOfRA-Preambles – 1) from the set of 64 preambles as defined in 36.211.

If Random Access Preambles group B exists, the thresholds, messagePowerOffsetGroupB and messageSizeGroupA, the configured UE transmitted power of the Serving Cell performing the Random Access Procedure, PCMAX, c [36.101], and the offset between the preamble and Msg3, deltaPreambleMsg3, that are required for selecting one of the two groups of Random Access Preambles (PCell only). // deltaPreambleMsg3 is power control related parameter (Refer to 36.331 and 36.213 5.1.1.1 UE behaviour for the details)

How to Generate 64 PRACH Preamble Sequences ?

As described above, the maximum number of PRACH Sequence a UE can use in a cell is 64. How these 64 different types of PRACH Sequence can be generated ? The procedure is as follows.

i) Generate a Zaddoff Chu sequence (849 samples) using rootSequenceIndex (let's call this sequence as 'base sequence')
ii) Generate 64 different sequency by doing cyclic shift of the base sequence. The cyclic shift interval is determined by Ncs and the Ncs is determined by zeroCorrelationZoneConfig and Highspeedflag.

For example >
Let's suppose SIB2 broadcast the parameters as follows.

a) rootSequenceindex = 22
b) Highspeedflag = false
c) zeroCorrelationZoneConfig = 5

From a), you will get the base Zaddoff-Chu sequence with u = 1 (Refer to rootSequenceIndex section if you want to know how this number is derived).
From b) and c), you will get the Ncs (Cyclicshift interval) = 26 (Refer to zeroCorrelationZoneConfig and Highspeedflag. section if you want to know how this number is derived).

Now you can get the 64 different PRACH sequence as follows.

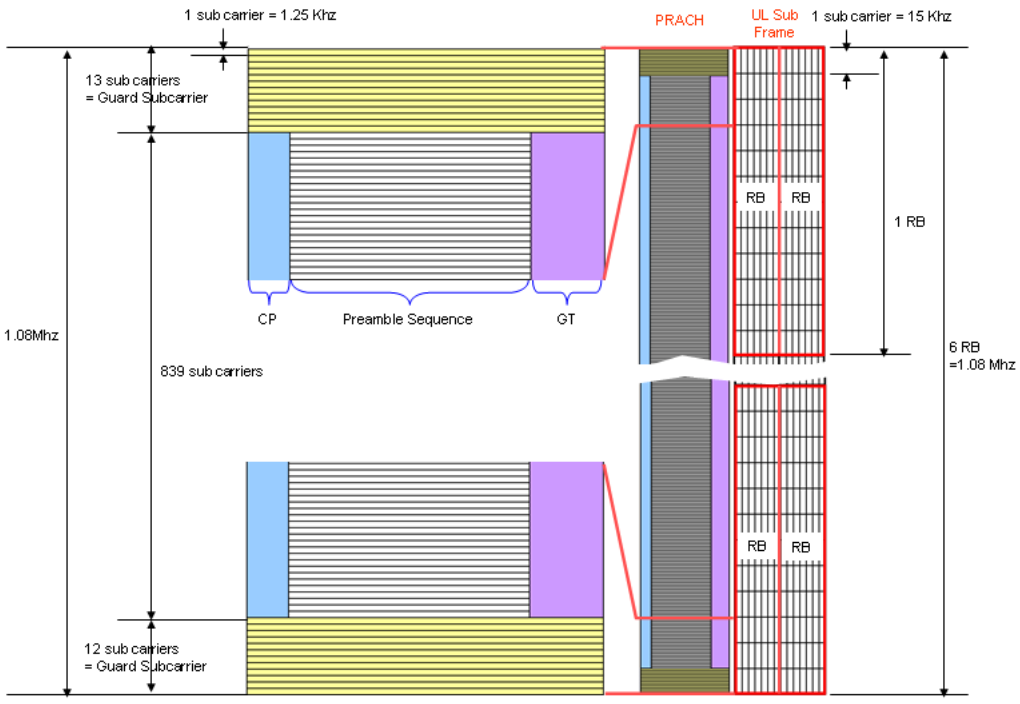
PRACH Sequence[0] = base sequence
PRACH Sequence[1] = do cyclic shift to base sequence by 1 * 26 samples
PRACH Sequence[2] = do cyclic shift to base sequence by 2 * 26 samples
....
PRACH Sequence[31] = do cyclic shift to base sequence by 31 * 26 samples
PRACH Sequence[32] = do cyclic shift to base sequence +1
PRACH Sequence[33] = do cyclic shift to base sequence +1 by 1 * 26 samples
PRACH Sequence[34] = do cyclic shift to base sequence +1 by 2 * 26 samples
.....
PRACH Sequence[63] = do cyclic shift to base sequence+1 by 31 * 26 samples

PRACH Signal Structure

Following figure shows the PRACH Premable signal structure in comparison with normal Uplink subframe. A couple of points to be specially mentioned are

- Preamble Length in Frequency Domain is amount to 6 RBs of UL Subframe, which is 1.08 Mhz
- Preamble Length in Time Domain including Guard Time (= CP Length + SEQUENCY Length + GT Length) can be 1 or 2 or 3 depending on Preamble Format
- One sub carrier of PRACH Preamble is 1.25 Khz whereas 1 sub carrier of UL subframe is 15 Khz. It means that 12 preamble sub carrier is amount to 1 UL Subframe subcarrier.

< PRACH Resource Grid >



How to generate RACH Signal ?

You don't have to know the details of this procedure unless you are the DSP or FPGA engineer implementing LTE PHY.

< PRACH Signal in Frequency Domain >

Just as a common sense about LTE, let's know that PRACH is a kind of ZaddOff Chu Sequence generated by the following equation. Be aware that this sequence is allocated along Frequency Domain.

$$x_u(n) = e^{-j\frac{\pi un(n+1)}{N_{ZC}}}, \quad 0 \le n \le N_{ZC}-1$$

, where u = physical root sequence index

UE can select a logical root sequence based on RachRootSequenceIndex. Once UE pick a specific Logical Root Sequence Index value, it can figure out the physical root sequence index (u) based on Table 5.7.2-4.

Nzc indicate 'number of data in the ZaddOff Chu Sequence'. This number is fixed to be 839 in preamble format 1,2,3 and 139 in preamble format 4.

There are 64 preambles available for each cell and UE has to be able to generate the 64 preambles for the cell it want to camp on.
You can easily generate 64 different preambles just by cyclically shifting an existing sequence, but there is a condition for this. All the preamle sequences should be othogonal to each other. Otherwise, various preambles from multiple UEs within the same cell can interfere each other. So we have to shift the generated sequence by a specifically designed value and this value is called Cv (Cyclic Shift Value) and it is defined as follows. (I think determining the Cv is one of the most complicated process in PRACH preamble generation because it gets involved with so many different parameters in cascading manner).

$$C_v = \begin{cases} vN_{CS} & v = 0,1,\ldots,\lfloor N_{ZC}/N_{CS} \rfloor - 1, N_{CS} \neq 0 \quad \text{for unrestricted sets} \\ 0 & N_{CS} = 0 \quad \text{for unrestricted sets} \\ d_{start}\left\lfloor v/n_{shift}^{RA} \right\rfloor + (v \bmod n_{shift}^{RA})N_{CS} & v = 0,1,\ldots,n_{shift}^{RA}n_{group}^{RA} + \bar{n}_{shift}^{RA} - 1 \quad \text{for restricted sets} \end{cases}$$

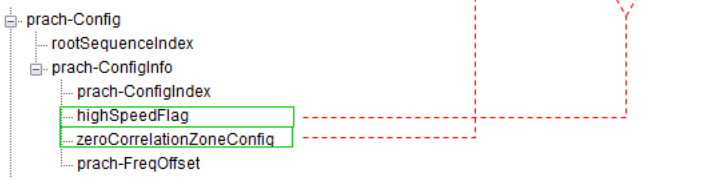
First, you would notice that we use different process to calculate Cv depending on whether we use 'unrestricted sets' or 'restricted sets'. This decision is made by 'Highspeedflag' information elements in SIB2. If Highspeedflag is set to be TRUE, we have to use 'restricted sets' and if Highspeedflag is false, we have to use 'unrestricted sets'.

N_cs is specified by zeroCorrelationZoneConfig information elements in SIB2. As you see in this mapping, N_cs values also gets different depending on whether we use 'restricted sets' or 'unrestricted sets'.

< Interpretation of highSpeedFlag / zeroCorrelationZoneConfig >

Table 5.7.2-2: N_{CS} for preamble generation (preamble formats 0-3).

N_{CS} configuration	N_{CS} value	
	Unrestricted set	Restricted set
0	0	15
1	13	18
2	15	22
3	18	26
4	22	32
5	26	38
6	32	46
7	38	55
8	46	68
9	59	82
10	76	100
11	93	128
12	119	158
13	167	202
14	279	237
15	419	-



Now let's look at how we get Nzc. This is pretty straightforward. Nzc is determined by the following table. (Preamble format 4 is used only in TDD LTE. So in case of FDD, you can say Nzc is fixed to be 839)

< 36.211-Table 5.7.2-1: Random access preamble sequence length >

Preamble format	N_{ZC}
0 – 3	839
4	139

If the Preamble is using the unrestricted sets, it is pretty simple. You only have to know Nzc, Ncs to figure out Cv. The problem is when the Preamble is using the 'restricted sets'. As you see the equation above, you need to know the following 4 values to figure out Cv in 'restricted sets'.

$$n_{shift}^{RA},\ d_{start},\ n_{group}^{RA},\ \bar{n}_{shift}^{RA}$$

The problem is that the calculation of these four variable is very complicated as shown below.

$$
\begin{cases}
n_{\text{shift}}^{\text{RA}} = \lfloor d_u / N_{\text{CS}} \rfloor \\
d_{\text{start}} = 2d_u + n_{\text{shift}}^{\text{RA}} N_{\text{CS}} \\
n_{\text{group}}^{\text{RA}} = \lfloor N_{\text{ZC}} / d_{\text{start}} \rfloor \\
\overline{n}_{\text{shift}}^{\text{RA}} = \max\left(\lfloor (N_{\text{ZC}} - 2d_u - n_{\text{group}}^{\text{RA}} d_{\text{start}}) / N_{\text{CS}} \rfloor, 0\right)
\end{cases}
\quad \text{For } N_{\text{CS}} \le d_u < N_{\text{ZC}}/3
$$

$$
\begin{cases}
n_{\text{shift}}^{\text{RA}} = \lfloor (N_{\text{ZC}} - 2d_u) / N_{\text{CS}} \rfloor \\
d_{\text{start}} = N_{\text{ZC}} - 2d_u + n_{\text{shift}}^{\text{RA}} N_{\text{CS}} \\
n_{\text{group}}^{\text{RA}} = \lfloor d_u / d_{\text{start}} \rfloor \\
\overline{n}_{\text{shift}}^{\text{RA}} = \min\left(\max\left(\lfloor (d_u - n_{\text{group}}^{\text{RA}} d_{\text{start}}) / N_{\text{CS}} \rfloor, 0\right), n_{\text{shift}}^{\text{RA}}\right)
\end{cases}
\quad \text{For } N_{\text{ZC}}/3 \le d_u \le (N_{\text{ZC}} - N_{\text{CS}})/2
$$

No cyclic shift — Except the two condition above

You would noticed that you need another value to calculate to determine which of the three case we have to use. It is du. So we need another process to determine du.

cyclic shift corresponding to a Doppler shift of magnitude $1/T_{\text{SEQ}}$

$$
d_u = \begin{cases} p & 0 \le p < N_{\text{ZC}}/2 \\ N_{\text{ZC}} - p & \text{otherwise} \end{cases}
$$

where p is the smallest non-negative integer that fulfils $(pu) \bmod N_{\text{ZC}} = 1$

We went through a complicated procedure just to determin one number (Cv). Once we get Cv, we can generate multiple preambles using the following function.

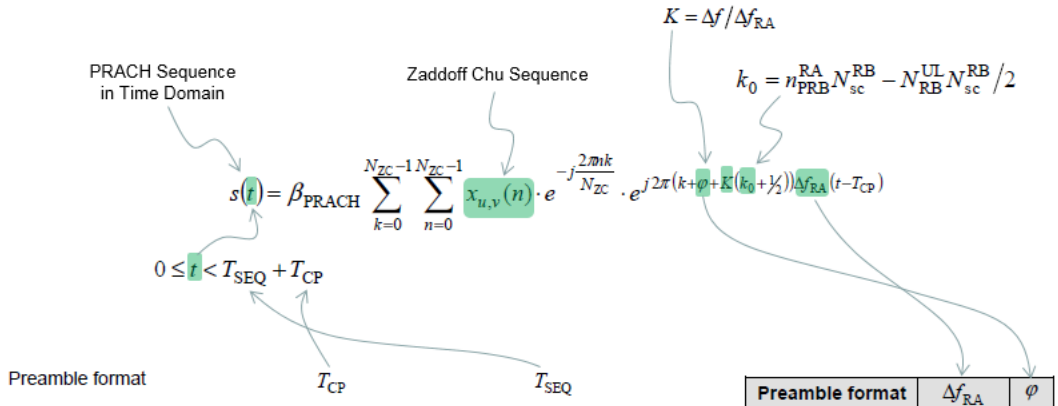
$$x_{u,v}(n) = x_u((n + C_v) \bmod N_{\text{ZC}})$$

< PRACH Signal in Time Domain >

Anyway, now we got a PRACH Preamble sequence in hand, but this is not all. In order to transmit this data. We have to convert this data into a time domain sequence and this conversion is done by the following process.

Once the frequency domain Zaddoff Chu sequence is obtained as described above, you can plug it into the following equation and with some additional parameters you can generate the time domain PRACH sequence.

< PRACH Signal Generation in Time Domain >



Preamble format	T_{CP}	T_{SEQ}
0	$3168 \cdot T_s$	$24576 \cdot T_s$
1	$21024 \cdot T_s$	$24576 \cdot T_s$
2	$6240 \cdot T_s$	$2 \cdot 24576 \cdot T_s$
3	$21024 \cdot T_s$	$2 \cdot 24576 \cdot T_s$
4*	$448 \cdot T_s$	$4096 \cdot T_s$

Preamble format	Δf_{RA}	φ
0 – 3	1250 Hz	7
4	7500 Hz	2

36.211 Table 5.7.3-1: Random access baseband parameters

< 36.211 Table 5.7.1-1: Random access preamble parameters >

Note : For the whole PRACH generation procedure, please refer to 5.7.2/5.7.3 of TS 36.211.

Exactly when and where Network transmit RACH Response

We all knows that Network should transmit RACH Response after it recieved RACH Preamble from UE, but do we know exactly when, in exactly which subframe, the network should transmit the RACH Response ? The following is what 3GPP 36.321 (section 5.1.4) describes.

Once the Random Access Preamble is transmitted and regardless of the possible occurrence of a measurement gap, the UE shall monitor the PDCCH for Random Access Response(s) identified by the RA-RNTI defined below, in the RA

Response window which starts at the subframe that contains the end of the preamble transmission [7] plus three subframes and has length ra-ResponseWindowSize subframes.

It means the earliest time when the network can transmit the RACH response is 3 subframe later from the end of RACH Preamble. Then what is the latest time when the network can transmit it ? It is determined by ra-ResponseWindowSize. This window size can be the number between 0 and 10 in the unit of subframes. This means that the maximum time difference between the end of RACH preamble and RACH Response is only 12 subframes (12 ms) which is pretty tight timing requirement.

How is the RACH Preamble Power determined ?

The RACH Preamble (PRACH) power varies depending on cases as described below (Based on 36.321 - 5.1.3) :
NOTE : In this section, I will describe on PRACH power only for Legacy LTE. The power varies a little for LTE BL/CE (LTE Cat M1) and LTE NB(Cat M2) which will be explained in separate page.
Basically PRACH power is determined by OpenLoopPower control algorithm. It would be good idea to read Open Loop and Closed Loop Power Control Page if you are not familiar with the concept.

Regardless of the cases, the PRACH Power (P_PRACH) is determined by the following equation.

P_PRACH = min{P_CMAX, PREAMBLE_RECEIVED_TARGET_POWER + PL}

PL stands for Path Loss between eNB Tx antenna and UE Rx Antenna. PREAMBLE_RECEIVED_TARGET_POWER is the PRACH power that eNB expect to recieve.

The meaning of this equation is as follows :

i) Calculate PREAMBLE_RECEIVED_TARGET_POWER + PL
ii) if the calculated value is less than P_CMAX(23 dBm), transmit the PRACH at the calculated value.
if the calculated value is greater than P_CMAX(23 dBm), transmit the PRACH at P_CMAX

Now let's think of how "PREAMBLE_RECEIVED_TARGET_POWER + PL" is calculated. This calculation varies depending on cases asdescribed below.

First let's see how PREAMBLE_RECEIVED_TARGET_POWER. This is calculated as follows.

< Case 1 > When UE send the first PRACH

PREAMBLE_RECEIVED_TARGET_POWER = preambleInitialReceivedTargetPower(in SIB2) + DELTA_PREAMBLE

< Case 2 > When UE retransmit PRACH

PREAMBLE_RECEIVED_TARGET_POWER = preambleInitialReceivedTargetPower(in SIB2) + DELTA_PREAMBLE
+ (PREAMBLE_TRANSMISSION_COUNTER – 1) * powerRampingStep

NOTE : As you see here, PREAMBLE_RECEIVED_TARGET_POWER will increase everytime it gets retransmitted
NOTE : PREAMBLE_TRANSMISSION_COUNTER starts from 1 at the first PRACH and get increased by 1 everytime PRACH get retransmitted.

Now you would ask on how to figure out preambleInitialReceivedTargetPower and powerRampingStep. These are infromed to UE in SIB2 as shown below.

```
+-sib2 ::= SEQUENCE [00]
  +-ac-BarringInfo ::= SEQUENCE OPTIONAL:Omit
  +-radioResourceConfigCommon ::= SEQUENCE
  | +-rach-Config ::= SEQUENCE
  | | +-preambleInfo ::= SEQUENCE [0]
  | | | +-numberOfRA-Preambles ::= ENUMERATED [n52]
  | | | +-preamblesGroupAConfig ::= SEQUENCE OPTIONAL:Omit
  | | +-powerRampingParameters ::= SEQUENCE
  | | | +-powerRampingStep ::= ENUMERATED [dB2]
  | | | +-preambleInitialReceivedTargetPower ::= ENUMERATED [dBm-104]
  | | +-ra-SupervisionInfo ::= SEQUENCE
  | | | +-preambleTransMax ::= ENUMERATED [n6]
  | | | +-ra-ResponseWindowSize ::= ENUMERATED [sf10]
  | | | +-mac-ContentionResolutionTimer ::= ENUMERATED [sf48]
```

You would still find one unknown parameter called DELTA_PREAMBLE. It is determined by which Preamble Format is used as shown below. Now you may ask how Preamble Format is determined. It is determined by prach-Configindex in SIB2.

< 36.321 - Table 7.6-1: DELTA_PREAMBLE values. >

Preamble Format	DELTA_PREAMBLE value
0	0 dB
1	0 dB
2	-3 dB
3	-3 dB
4	8 dB

Now let's think of how UE can estimate the path loss between eNB Tx antenna and UE Rx antenna. Technically it is simple. It can be calculated as follows.

PL = eNB Transmitter Power - UE Reciever Power

UE has 'UE Reciever Power' by direct measurement. The problem is how UE can figure out eNB transmitter Power. It is not possible for UE to figure this out direct measurement. eNB should tell UE of its transmitted power. This eNB Tx power is notifed by a SIB2 parameter referenceSignalPower as shown below.

```
+-sib2 ::= SEQUENCE [00]
  +-ac-BarringInfo ::= SEQUENCE OPTIONAL:Omit
  +-radioResourceConfigCommon ::= SEQUENCE
  | +-rach-Config ::= SEQUENCE
  | | +-preambleInfo ::= SEQUENCE [0]
  | | | +-numberOfRA-Preambles ::= ENUMERATED [n52]
  | | | +-preamblesGroupAConfig ::= SEQUENCE OPTIONAL:Omit
  | | +-powerRampingParameters ::= SEQUENCE
  | | | +-powerRampingStep ::= ENUMERATED [dB2]
  | | | +-preambleInitialReceivedTargetPower ::= ENUMERATED [dBm-104]
  | | +-ra-SupervisionInfo ::= SEQUENCE
  | | | +-preambleTransMax ::= ENUMERATED [n6]
  | | | +-ra-ResponseWindowSize ::= ENUMERATED [sf10]
  | | | +-mac-ContentionResolutionTimer ::= ENUMERATED [sf48]
  | | +-maxHARQ-Msg3Tx ::= INTEGER (1..8) [4]
  | +-bcch-Config ::= SEQUENCE
  | +-pcch-Config ::= SEQUENCE
  | +-prach-Config ::= SEQUENCE
  | +-pdsch-Config ::= SEQUENCE
  | | +-referenceSignalPower ::= INTEGER (-60..50) [18]
  | | +-p-b ::= INTEGER (0..3) [0]
```

Example >

Let's take an example case and apply all the stuffs described above. Let's assume that UE decoded following SIB parameters.

- preambleInitialReceivedTargetPower = -104 (dBm)
- referenceSignalPower = 18
- prach-ConfigIndex: 3 ==> You can figure out that PREAMBLE FORMAT is Format 0.
- powerRampingStep = dB2

Now assume that UE measures RSRP at its reciever antenna as follows.

- RSRP measuredt at UE = -80dBm

From these information, you can calculate the following value

PREAMBLE_RECEIVED_TARGET_POWER = preambleInitialReceivedTargetPower(in SIB2) + DELTA_PREAMBLE
= -104 + 0 = -104

Now you can calculate the PathLoss as below.

PL = (referenceSignalPowerin SIB2) - (RSRP measuredt at UE) = 18 - (-80) = 98

Then just plug these into PRACH power equation and you get the result as follows.

P_PRACH = min{23, -104+98 } = min(23, -6) = 6 dBm

PRACH Parameters and Physical Meaning

< prach-ConfigIndex >

This parameter determines what type of preamble format should be used and at which system frame and subframe UE can transmit PRACH Preamble.

```
sib2
   radioResourceConfigCommon
      rach-ConfigCommon
         preambleInfo
            numberOfRA-Preambles: n52 (12)
         powerRampingParameters
            powerRampingStep: dB2 (1)
            preambleInitialReceivedTargetPower: dBm-104 (8)
         ra-SupervisionInfo
            preambleTransMax: n6 (3)
            ra-ResponseWindowSize: sf10 (7)
            mac-ContentionResolutionTimer: sf48 (5)
         maxHARQ-Msg3Tx: 4
      ...
      prach-Config
         rootSequenceIndex: 22
         prach-ConfigInfo
            prach-ConfigIndex: 3
            ..0. .... highSpeedFlag: False
            zeroCorrelationZoneConfig: 5
            prach-FreqOffset: 4
      ...

      additionalSpectrumEmission: 1
   timeAlignmentTimerCommon: infinity (7)
```

The meaning of prach-ConfigIndex is defined by the following table. If you are not familiar with how to interpret this table. Refer to the section Exactly when and Where a UE transmit RACH ?

< SIB 2 and 36.211 Table 5.7.1-2: Frame structure type 1 random access configuration for preamble format 0-3.>

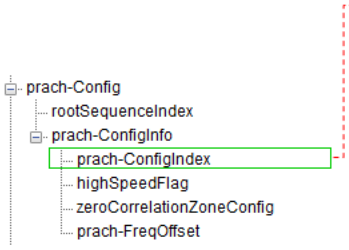


Table 5.7.1-2: Frame structure type 1 random access configuration for preamble format 0-3.

PRACH Configuration Index	Preamble Format	System frame number	Subframe number	PRACH Configuration Index	Preamble Format	System frame number	Subframe number
0	0	Even	1	32	2	Even	1
1	0	Even	4	33	2	Even	4
2	0	Even	7	34	2	Even	7
3	0	Any	1	35	2	Any	1
4	0	Any	4	36	2	Any	4
5	0	Any	7	37	2	Any	7
6	0	Any	1, 6	38	2	Any	1, 6
7	0	Any	2 ,7	39	2	Any	2 ,7
8	0	Any	3, 8	40	2	Any	3, 8
9	0	Any	1, 4, 7	41	2	Any	1, 4, 7
10	0	Any	2, 5, 8	42	2	Any	2, 5, 8
11	0	Any	3, 6, 9	43	2	Any	3, 6, 9
12	0	Any	0, 2, 4, 6, 8	44	2	Any	0, 2, 4, 6, 8
13	0	Any	1, 3, 5, 7, 9	45	2	Any	1, 3, 5, 7, 9
14	0	Any	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	46	N/A	N/A	N/A
15	0	Even	9	47	2	Even	9
16	1	Even	1	48	3	Even	1
17	1	Even	4	49	3	Even	4
18	1	Even	7	50	3	Even	7
19	1	Any	1	51	3	Any	1
20	1	Any	4	52	3	Any	4
21	1	Any	7	53	3	Any	7
22	1	Any	1, 6	54	3	Any	1, 6
23	1	Any	2 ,7	55	3	Any	2 ,7
24	1	Any	3, 8	56	3	Any	3, 8
25	1	Any	1, 4, 7	57	3	Any	1, 4, 7
26	1	Any	2, 5, 8	58	3	Any	2, 5, 8
27	1	Any	3, 6, 9	59	3	Any	3, 6, 9
28	1	Any	0, 2, 4, 6, 8	60	N/A	N/A	N/A
29	1	Any	1, 3, 5, 7, 9	61	N/A	N/A	N/A
30	N/A	N/A	N/A	62	N/A	N/A	N/A
31	1	Even	9	63	3	Even	9

< zeroCorrelationZoneConfig and Highspeedflag >

zeroCorrelationZoneConfig and Highspeedflg Ie is to specify the cyclic shift intervals to generate 64 PRACH Sequence from a single base sequence. These IE(information elements) are specified in SIB2 as in the example below.

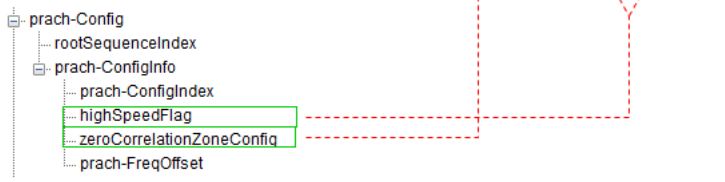
```
sib2
   radioResourceConfigCommon
      rach-ConfigCommon
         preambleInfo
            numberOfRA-Preambles: n52 (12)
         powerRampingParameters
            powerRampingStep: dB2 (1)
            preambleInitialReceivedTargetPower: dBm-104 (8)
         ra-SupervisionInfo
            preambleTransMax: n6 (3)
            ra-ResponseWindowSize: sf10 (7)
            mac-ContentionResolutionTimer: sf48 (5)
         maxHARQ-Msg3Tx: 4
      ...
      prach-Config
         rootSequenceIndex: 22
         prach-ConfigInfo
            prach-ConfigIndex: 3
            ..0. .... highSpeedFlag: False
            zeroCorrelationZoneConfig: 5
            prach-FreqOffset: 4
      ...

      additionalSpectrumEmission: 1
   timeAlignmentTimerCommon: infinity (7)
```

If you apply the values in the example above, you would get the Ncs value of 26. (Pick the number at the cross of column 'Unrestricted Set (HighSpeedFlack = False) and the row of Ncs 5)

Table 5.7.2-2: N_{CS} for preamble generation (preamble formats 0-3).

N_{CS} configuration	N_{CS} value	
	Unrestricted set	Restricted set
0	0	15
1	13	18
2	15	22
3	18	26
4	22	32
5	26	38
6	32	46
7	38	55
8	46	68
9	59	82
10	76	100
11	93	128
12	119	158
13	167	202
14	279	237
15	419	-



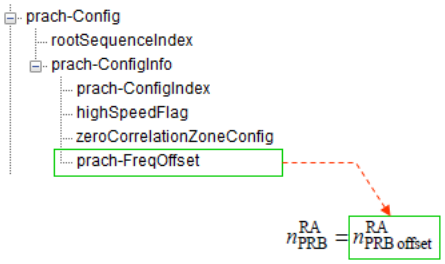
NOTE : If you want to check on which spec defines the mapping between highSpeedFlag and Ncs value set, see 36.331. It states as follows.

highSpeedFlag : TRUE corresponds to Restricted set and FALSE to Unrestricted set

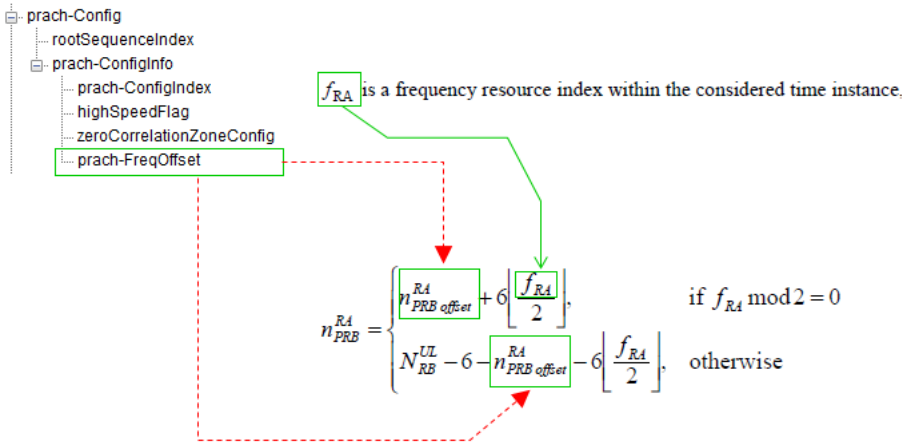
< prach-FreqOffset >

prach-FreqOffset is the parameter that determines the location of PRACH preamble in frequency domain. This location in frequency domain is calculated in the unit of PRM index and calulcated by following equation. As you see, the equation gets different depending on Preamble Format.

< FDD : PRACH Preamble Location for Preamble Format 0-4>



< TDD : PRACH Preamble Location for Preamble Format 0-3 >



< rootSequenceIndex >

there are 838 root Zadoff-Chu sequences available for preambles. The length of each root sequence is 839. RootConfigurationIndex informs the UE on which sequence to use via SIB2 as shown below.

```
sib2
   radioResourceConfigCommon
      rach-ConfigCommon
         preambleInfo
            numberOfRA-Preambles: n52 (12)
         powerRampingParameters
            powerRampingStep: dB2 (1)
            preambleInitialReceivedTargetPower: dBm-104 (8)
         ra-SupervisionInfo
            preambleTransMax: n6 (3)
            ra-ResponseWindowSize: sf10 (7)
            mac-ContentionResolutionTimer: sf48 (5)
         maxHARQ-Msg3Tx: 4
      ...
      prach-Config
         rootSequenceIndex: 22
         prach-ConfigInfo
            prach-ConfigIndex: 3
            ..0. .... highSpeedFlag: False
            zeroCorrelationZoneConfig: 5
            prach-FreqOffset: 4
      ...

      additionalSpectrumEmission: 1
   timeAlignmentTimerCommon: infinity (7)
```

This rootSequenceIndex is a logical value. The real number (physical number) is called 'u' which is a variable used to generate PRACH ZaddOff-Chu Sequence. The mapping between the rootsequenceIndex and 'u' is determined by the following table. For example, if the rootsequenceIndex is 22 as in the example shown above, the 'u' become '1' according to this table.

Example 1 >

You may wonder how the 'rootsequenceIndex = 22' is translated to 'u = 1'. It is simple. From the Table 5.7.2-4, you can pick the Logical root sequence number range that the rootsequenceIndex belong to. For example, the range that 22 belong to is as follows.

Logical root sequence number	Physical root sequence number u
0-23	129, 710, 140, 699, 120, 719, 210, 629, 168, 671, 84, 755, 105, 734, 93, 746, 70, 769, 60, 779,2, 837, 1, 838

Then you can convert the above row into a table as shown below and you can easily figure out the 'roolSequenceIndex=22' is mapped to 'u=1'.

Logical #	0	1	2	3	4	5	6	7	8	9	10	11
u	129	710	140	699	120	719	210	629	168	671	84	755
Logical #	12	13	14	15	16	17	18	19	20	21	22	23
u	105	734	93	746	70	769	60	779	2	837	1	838

Example 2 >

Just for another practice, let me give you another example. What would be the 'u' for the rootSequenceIndex=696 ? The answer is as follows :

Logical root sequence number	Physical root sequence number u
660–707	257, 582, 273, 566, 255, 584, 254, 585, 245, 594, 251, 588, 412, 427, 372, 467, 282, 557, 403, 436, 396, 443, 392, 447, 391, 448, 382, 457, 389, 450, 294, 545, 297, 542, 311, 528, 344, 495, 345, 494, 318, 521, 331, 508, 325, 514, 321, 518

Then you can convert the above row into a table as shown below and you can easily figure out the 'rootSequenceIndex=606' is mapped to 'u=344'.

Logical #	660	661	662	663	664	665	666	667	668	669	670	671
u	257	582	273	566	255	584	254	585	245	594	251	588
Logical #	672	673	674	675	676	677	678	679	680	681	682	683
u	412	427	372	467	282	557	403	436	396	443	392	447
Logical #	684	685	686	687	688	689	690	691	692	693	694	695
u	391	448	382	457	389	450	294	545	297	542	311	528
Logical #	696	697	698	699	700	701	702	703	704	705	706	707
u	344	495	345	494	318	521	331	508	325	514	321	518

< 36.211 Table 5.7.2-4: Root Zadoff-Chu sequence order for preamble formats 0 – 3 >

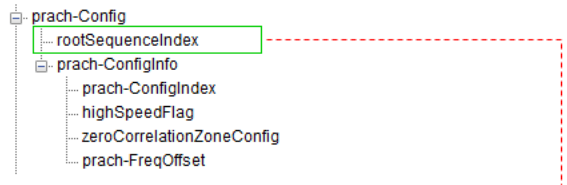


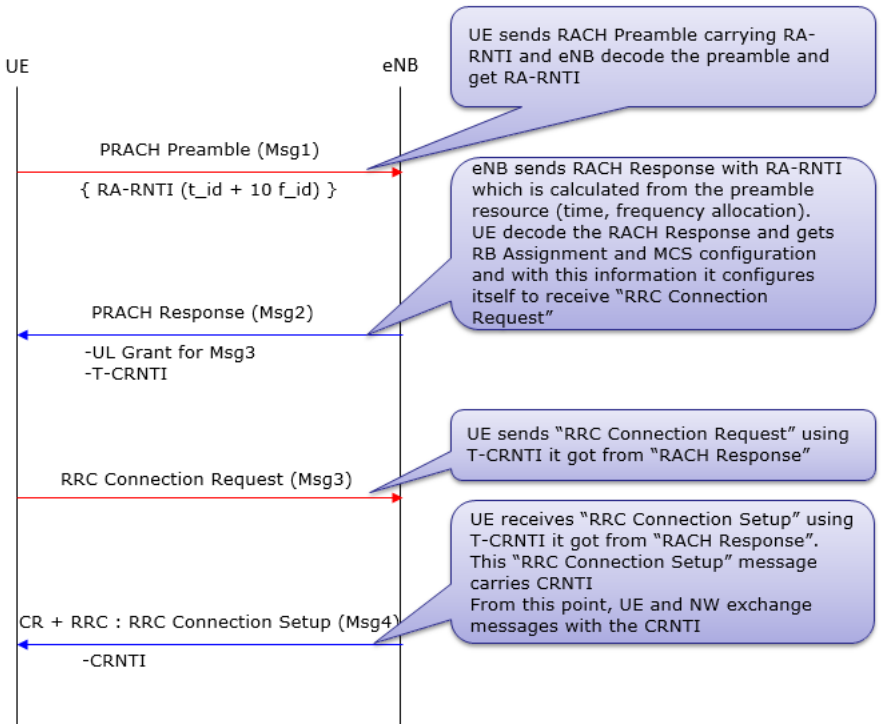
Table 5.7.2-4: Root Zadoff-Chu sequence order for preamble formats 0 – 3.

Logical root sequence number	Physical root sequence number u (in increasing order of the corresponding logical sequence number)
0–23	129, 710, 140, 699, 120, 719, 210, 629, 168, 671, 84, 755, 105, 734, 93, 746, 70, 769, 60, 779 2, 837, 1, 838
24–29	56, 783, 112, 727, 148, 691
30–35	80, 759, 42, 797, 40, 799
36–41	35, 804, 73, 766, 146, 693
42–51	31, 808, 28, 811, 30, 809, 27, 812, 29, 810
52–63	24, 815, 48, 791, 68, 771, 74, 765, 178, 661, 136, 703
64–75	86, 753, 78, 761, 43, 796, 39, 800, 20, 819, 21, 818
76–89	95, 744, 202, 637, 190, 649, 181, 658, 137, 702, 125, 714, 151, 688
90–115	217, 622, 128, 711, 142, 697, 122, 717, 203, 636, 118, 721, 110, 729, 89, 750, 103, 736, 61, 778, 55, 784, 15, 824, 14, 825
116–135	12, 827, 23, 816, 34, 805, 37, 802, 46, 793, 207, 632, 179, 660, 145, 694, 130, 709, 223, 616
136–167	228, 611, 227, 612, 132, 707, 133, 706, 143, 696, 135, 704, 161, 678, 201, 638, 173, 666, 106, 733, 83, 756, 91, 748, 66, 773, 53, 786, 10, 829, 9, 830
168–203	7, 832, 8, 831, 16, 823, 47, 792, 64, 775, 57, 782, 104, 735, 101, 738, 108, 731, 208, 631, 184, 655, 197, 642, 191, 648, 121, 718, 141, 698, 149, 690, 216, 623, 218, 621
204–263	152, 687, 144, 695, 134, 705, 138, 701, 199, 640, 162, 677, 176, 663, 119, 720, 158, 681, 164, 675, 174, 665, 171, 668, 170, 669, 87, 752, 169, 670, 88, 751, 107, 732, 81, 758, 82, 757, 100, 739, 98, 741, 71, 768, 59, 780, 65, 774, 50, 789, 49, 790, 26, 813, 17, 822, 13, 826, 6, 833
264–327	5, 834, 33, 806, 51, 788, 75, 764, 99, 740, 96, 743, 97, 742, 166, 673, 172, 667, 175, 664, 187, 652, 163, 676, 185, 654, 200, 639, 114, 725, 189, 650, 115, 724, 194, 645, 195, 644, 192, 647, 182, 657, 157, 682, 156, 683, 211, 628, 154, 685, 123, 716, 139, 700, 212, 627, 153, 686, 213, 626, 215, 624, 150, 689
328–383	225, 614, 224, 615, 221, 618, 220, 619, 127, 712, 147, 692, 124, 715, 193, 646, 205, 634, 206, 633, 116, 723, 160, 679, 186, 653, 167, 672, 79, 760, 85, 754, 77, 762, 92, 747, 58, 781, 62, 777, 69, 770, 54, 785, 36, 803, 32, 807, 25, 814, 18, 821, 11, 828, 4, 835
384–455	3, 836, 19, 820, 22, 817, 41, 798, 38, 801, 44, 795, 52, 787, 45, 794, 63, 776, 67, 772, 72 767, 76, 763, 94, 745, 102, 737, 90, 749, 109, 730, 165, 674, 111, 728, 209, 630, 204, 635, 117, 722, 188, 651, 159, 680, 198, 641, 113, 726, 183, 656, 180, 659, 177, 662, 196, 643, 155, 684, 214, 625, 126, 713, 131, 708, 219, 620, 222, 617, 226, 613
456–513	230, 609, 232, 607, 262, 577, 252, 587, 418, 421, 416, 423, 413, 426, 411, 428, 376, 463, 395, 444, 283, 556, 285, 554, 379, 460, 390, 449, 363, 476, 384, 455, 388, 451, 386, 453, 361, 478, 387, 452, 360, 479, 310, 529, 354, 485, 328, 511, 315, 524, 337, 502, 349, 490, 335, 504, 324, 515
514–561	323, 516, 320, 519, 334, 505, 359, 480, 295, 544, 385, 454, 292, 547, 291, 548, 381, 458, 309, 440, 380, 459, 397, 442, 369, 470, 377, 462, 410, 429, 407, 432, 281, 558, 414, 425, 247, 592, 277, 562, 271, 568, 272, 567, 264, 575, 259, 580
562–629	237, 602, 239, 600, 244, 595, 243, 596, 275, 564, 278, 561, 250, 589, 246, 593, 417, 422, 248, 591, 394, 445, 393, 446, 370, 469, 365, 474, 300, 539, 299, 540, 364, 475, 362, 477, 298, 541, 312, 527, 313, 526, 314, 525, 353, 486, 352, 487, 343, 496, 327, 512, 350, 489, 326, 513, 319, 520, 332, 507, 333, 506, 348, 491, 347, 492, 322, 517
630–659	330, 509, 338, 501, 341, 498, 340, 499, 342, 497, 301, 538, 366, 473, 401, 438, 371, 468, 408, 431, 375, 464, 249, 590, 269, 570, 238, 601, 234, 605
660–707	257, 582, 273, 566, 255, 584, 254, 585, 245, 594, 251, 588, 412, 427, 372, 467, 282, 557, 403, 436, 396, 443, 392, 447, 391, 448, 382, 457, 389, 450, 294, 545, 297, 542, 311, 528, 344, 495, 345, 494, 318, 521, 331, 508, 325, 514, 321, 518
708–729	346, 493, 339, 500, 351, 488, 306, 533, 289, 550, 400, 439, 378, 461, 374, 465, 415, 424, 270, 569, 241, 598
730–751	231, 608, 260, 579, 268, 571, 276, 563, 409, 430, 398, 441, 290, 549, 304, 535, 308, 531, 358, 481, 316, 523
752–765	293, 546, 288, 551, 284, 555, 368, 471, 253, 586, 256, 583, 263, 576
766–777	242, 597, 274, 565, 402, 437, 383, 456, 357, 482, 329, 510
778–789	317, 522, 307, 532, 286, 553, 287, 552, 266, 573, 261, 578
790–795	236, 603, 303, 536, 356, 483
796–803	355, 484, 405, 434, 404, 435, 406, 433
804–809	235, 604, 267, 572, 302, 537
810–815	309, 530, 265, 574, 233, 606
816–819	367, 472, 296, 543
820–837	336, 503, 305, 534, 373, 466, 280, 559, 279, 560, 419, 420, 240, 599, 258, 581, 229, 610

RACH Procedure during Initial Registration - RACH Procedure Summary

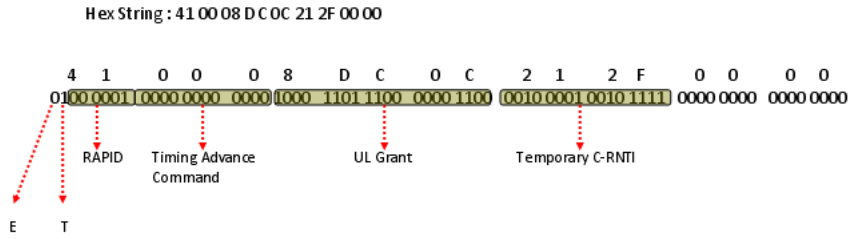
Follwing is an example of RACH procedure which happens during the initiail registration. If you will be an engineer who is working on protocol stack development or test case development, you should be very familiar with all the details of this process.

< RACH Procedure during Initial Registration >



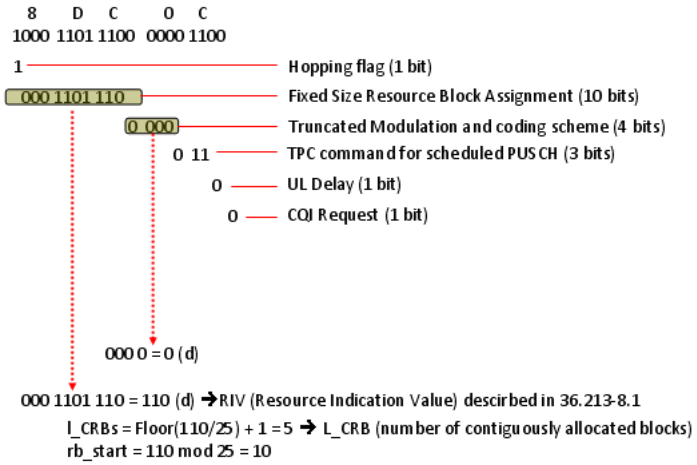
Again, we have to know every details of every step without missing anything to be a developer, but of course it is not easy to understand everything at a single shot. So, let's start with something the most important part, which I think is the details of RACH response. Following diagram shows one example of RACH Response with 5 Mhz bandwidth. We don't have to memorize the detailed value itself but should be familiar with the data format and understand which part of this bit string means what.

< An Example of RAR (RACH Response) >



If you decode UL Grant part, you will get the following result. You will notice that the information it carries would be very similar to DCI format 0 which carries Resource Allocation for uplink data. This information in UL Grant in RACH Response message is the resource allocation for msg3 (e.g, RRC Connection Request). Note : This is example of RAR for System BW 5 Mhz. If the sytem BW gets different, you should have different RIV values (if you want to have the same Start_RB, N_RB as in this example) or you will have different Start_RB, N_RB (if you keep RIV as below and just change the system BW)

< Example of UL Grant in RAR >



Let me describe this procedure in verbal form again.

i) UE initiate a Random Access Procedure on the (uplink) Random Access Channel (RACH).(The location of RACH in the frequency/time resource grid the RACH is known to the mobile via the (downlink) Broadcast Channel (BCH))

ii) Network sends a Random Access Response Message(RARM) at a time and location on the Physical Downlink Shared Channel (PDSCH) (The time and location of RARM on PDSCH can be calculated from the time and location the random access message was sent. This message contains the random identity sent by the device, a Cell Radio Network Temporary ID (T_C-RNTI) which will be used for all further bandwidth assignments, and an initial uplink bandwidth assignment)

iii) The mobile device then uses the bandwidth assignment to send a short (around 80 bits) RRC Connection Request message which includes it's identity which has previously been assigned to it by the core network

Only the step i) uses physical-layer processing specifically designed for random access. The remaining steps utilizes the same physical layer processing as used for normal uplink and downlink data transmission

How can we get RA RNTI ?

"5.1.4 Random Access Response reception" in "TS36.321" says how to calculate RA_RNTI as follows.

The RA-RNTI associated with the PRACH in which the Random Access Preamble is transmitted, is computed as:
*RA-RNTI = 1 + t_id + 10 * f_id*
Where t_id is the index of the first subframe of the specified PRACH (0 <≤ t_id <10), and f_id is the index of the specified PRACH within that subframe, in ascending order of frequency domain (0≤ f_id< 6).

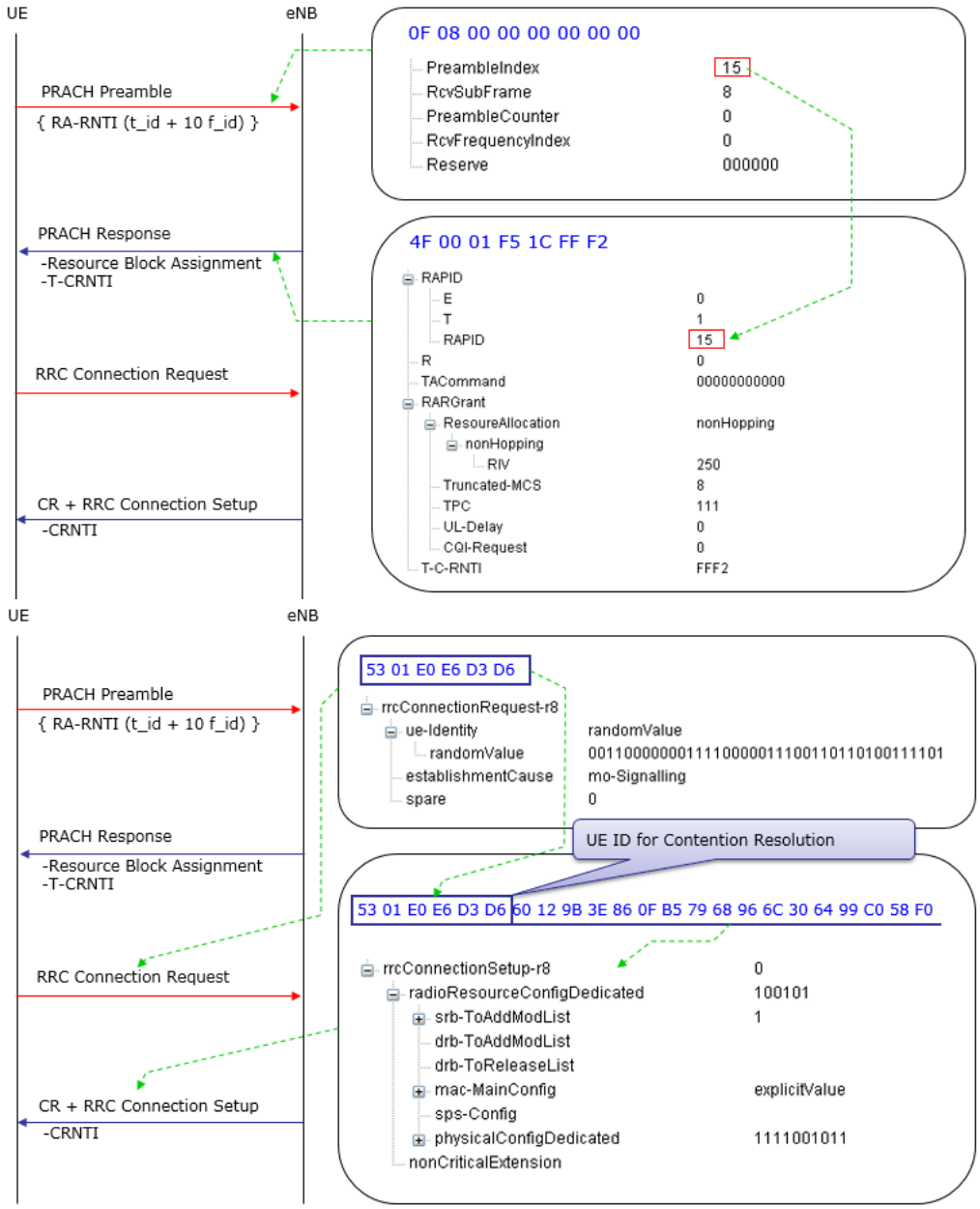
For FDD, f_id is fixed as 0.

Therefore, RA_RNTI is decided by the sending timing (SubFrame) of PRACH Preamble by UE. It means that (the subframe number (number between 0000~0009) of PRACH transmission + 1) is RA-RNTI.
It means that UE specifies RA_RNTI by the sending timing (SubFrame) of PRACH Preamble.

An Example of Full RACH Process

Following is an example of Full RACH process with a commercialized LTE device and LTE Network Emulator. I would not explain anything in detail. Just check if the following diagram make any sense to you. If it does, I would say you understand all the details that I explained above.

< RACH Sequence during Initial Attach >



Following is an example of real RACH procedure happening between a UE and Amarisoft OTS. Log shown here is a screencapture of Amarisoft Web Logging tool.

< Example of RACH Process during Initial Attach >

ENB	SFN	RNTI	Info	Message
PHY	154.5		PRACH	sequence_index=28 ta=0 snr=28.8 dB 1
MAC				Allocating new UE
MAC				RAR 2
PHY	155.2	0x5	PDSCH	harq=si type=2 rb_start=23 l_crb=2 CW0: tb_len=7 mod=2 rv_idx=0
PHY	155.2	0x5	PDCCH	cce_index=0/20 L=4 dci=1a
PHY	155.8	0x66	PUSCH	harq=6 type=0 rb_start=2 l_crb=4 CW0: tb_len=11 mod=2 rv_idx=0 crc=OK snr=31.3
MAC				LCID:0 len=6 PAD: len=2
RRC			CCCH	RRC Connection Request 3
RRC			CCCH	RRC Connection Setup
PHY	156.2	0x66	PHICH	hi=1 n_phich_group=2 n_phich_seq=0
MAC				PAD: len=0 UECRI: 40 10 00 00 00 14 LCID:0 len=23 4
PHY	156.2	0x66	PDSCH	harq=0 type=2 rb_start=19 l_crb=6 tx=port0 CW0: tb_len=32 mod=2 rv_idx=0
PHY	156.2	0x66	PDCCH	cce_index=12/20 L=4 dci=1a
PHY	156.6		PUCCH	format=1A n=24 snr=19.1 dB ack=1
PHY	160.4		PUCCH	format=1 n=0 sr=1 snr=19.5 dB
PHY	160.8	0x66	PDCCH	cce_index=16/20 L=4 dci=0
PHY	161.2	0x66	PUSCH	harq=4 type=0 rb_start=2 l_crb=1 CW0: tb_len=42 mod=4 rv_idx=0 crc=OK snr=16.4
MAC				PHR: ph=56 SBSR: lcg=0 b=0 LCID:1 len=14 PAD: len=21
RRC			DCCH	RRC Connection Setup Complete

PRACH Retransmission

Most part of previous section was about the ideal RACH process, which means that UE send PRACH and Network send RACH Response at the first trial and went through all the way to the end of process at the first trial.

What if UE does not receive RACH Response at the first trial ? What is UE supposed to do in this case ?
The answer is simple. Just retry (resend) PRACH. (In this case, UE might not have any Backoff Indicator value which normally transmitted in MAC CE being sent with RAR).

There is another case where UE needs to retry PRACH. It is the case where UE received RAR from the network, but the RAPID is not for it (It means that RAR is not for some other UE). In this case, it is highly probable that a Backoff Indicator value is transmitted with RAR to control the PRACH retransmission timing.

Then you would have more question. ("I" in the following description is "UE")

> i) When do I have to retry ? (What should be the time delay between the previous transmission and the next transmission ?)
> ii) Do I have to retransmit the PRACH with the same power as previous one ? Or try with a little bit higher power ? If I have to try with a little bit higher power, how much power do I have to increase ?
> iii) If I keep failing to receive RACH response, how many time I have to retry ? Do I have to retry until the battery runs out ? or retry only several times and give up ? If I have to give up after a certain amount of retry, exactly how many times do I have to retry ?

The answers to all of these questions are provided by the network.
The answer (instruction) to question i) is provided by Network via a special RAR MAC PDU called "Backoff Indicator".
The answer to question ii) and iii) are provided by Network via SIB2 as follows. powerRampingStep is the answer to question ii) and preambleTransMax is the answer to question iii).
In the following example, powerRampingStep = dB2. It means UE has to increase PRACH power by 2 dB everytime it retries.
preambleTransMax = n6. It means UE retries PRACH retransmit only 6 times and then give up. (This is my understanding at least as of now. But trying with real device, I see many cases UE does not give up even after it reaches preambleTransMax. I will get this updated as I find more)

```
|     +-radioResourceConfigCommon ::= SEQUENCE
|     | +-rach-Config ::= SEQUENCE
|     | | +-preambleInfo ::= SEQUENCE [0]
|     | | | +-numberOfRA-Preambles ::= ENUMERATED [n52]
|     | | | +-preamblesGroupAConfig ::= SEQUENCE OPTIONAL:Omit
|     | | +-powerRampingParameters ::= SEQUENCE
|     | | | +-powerRampingStep ::= ENUMERATED [dB2]
|     | | | +-preambleInitialReceivedTargetPower ::= ENUMERATED [dBm-104]
|     | | +-ra-SupervisionInfo ::= SEQUENCE
|     | | | +-preambleTransMax ::= ENUMERATED [n6]
|     | | | +-ra-ResponseWindowSize ::= ENUMERATED [sf10]
|     | | | +-mac-ContentionResolutionTimer ::= ENUMERATED [sf48]
|     | | +-maxHARQ-Msg3Tx ::= INTEGER (1..8) [4]
```

Additional Factors :
- PRACH Config Index (in SIB2)
- Backoff Indicator (in MAC CE)
- T-300 (in SIB2)

Following is an example of PRACH Retry being observed in a real device. This is the case where UE send PRACH and NW does not send RAR (Yellow cell indicates the timing determined by PRACH Config Index when UE is allowed to send PRACH. See Exactly when and where Network transmit RACH Response . Green cell indicates the timing when UE send PRACH in this specific example)

< RACH Retry >

SFN	Subframe	PRACH Allowed	PRACH Txed
849	0		
	1		
	2		PRACH 1
	3		min prep time
	4		
	5		
	6		ra-response WindowSize
	7		
	8		
	9		
850	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		PRACH 2
	9		min prep time
851	0		
	1		

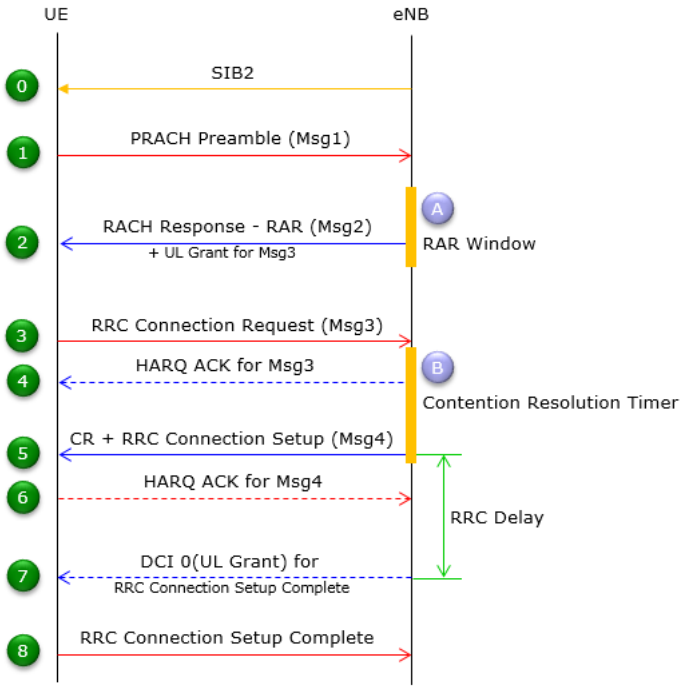
RACH Process Overview In Diagrams

I have explained long about the RACH process. Now you may ask "What is the trigger that let UE initiate the RACH process ?". You will see various triggers in 3GTS 36.300 (10.1.5) : Overall description of RACH Process. "Turning on UE" is one of the trigger for sure. And following is another trigger for this process.

< RACH Procedure on Initial Registration >

This is basically the same sequence that I explained in previous sections, but I simplified the diagram in previous sections to let reader focused more on messaging part of RACH procedure. In this diagram, you see some additional steps like HARQ ACK, DCI 0 (UL Grant). This flow is more similar to real live network procedure.

< RACH Procedure during Initial Registration >



Following is one example for this sequence that I got from live network and summarized with important parameters. I hope this can be a good practice for you. (Note : This is with FDD)

< Live network Example of RACH process for Initial Attach >

SFN : 402.4	RACH Preamble
	RNTI = None Timing Offset = 2 Logical Root = 219 Preamble index = 33 NC Configuration = 12

	Set Type = Unrestricted Logical Root = 215 Preamble Format = 0 RbStart = 2
SFN : 402.8	DCI Format 1 + MAC RA Response
	CCE Start = 0 CCE Length = 4 DCI Format 1A // Masked with RA_RNTI Format = 1 Distributed VRB flag = 0 (Local) Resource Allocation = 500 (RB Start = 0, RB Length = 11) MCS = 1 (I_TBS = 1) NDI (New Data Indicator) = 1 (True) RV = 0 MAC : 61 00 B0 C0 4C 2C 09 // RA-Response E = 0(False) T = 1 RAPID = 33 Timing Advanced = 11 Hopping Flag 0 = False Fixed Size Resource Block Assignment = 96 (RB Start = 46, RB Length = 2) MCS = 2, I_TBS = 2, rv = 0 TPC Command for PUCCH 3 = 0 UL Delay 0 = False CQI Request = False T_CRNTI = 11273
SFN : 403.4	PUSCH - RRC Connection Request
	MAC : 20 06 1F 5C 2C 04 B2 AC F6 Sub Header 0 R = OK E = 1 LCID = 0 (CCCH) F = 0 (False) L = 6 Sub Header 1 R = OK E = 0 LCID = 31 (Padding) CCCH-RLC : 5C 2C 04 B2 AC F6 (RRC Connection Request)
SFN : 403.8	PHICH-ACK
SFN : 404.7	PDCCH (DCI Format 1) + PDSCH (RRC Connection Setup)
	CCE Start = 0 CCE Length = 8 DCI Format 1A (Hex : 47D01E2) Format = 1 Distributed VRB flag = 0 (Local) Resource Allocation = 500 (RB Start = 0, RB Length = 11) MCS = 0 (I_TBS = 0) HARQ Process Number = 7 NDI (New Data Indicator) = 1 (True) RV = 0 TPC Command for PUCCH = 1 MAC : 3C 20 1A 1F 5C 2C 04 B2 AC F6 60 12 98 08 FD 4E ..... Sub Header 0 R = OK E = 1 LCID = 28 (UE Contention Resolution Identity) Sub Header 1 R = OK E = 1 LCID = 1 (CCCH) F = 0 (False) L = 26 Sub Header 2

	R = OK E = 0 LCID = 31 (Padding) UE Contention Resolution Identity UE Contention Resolution Identity= 5C 2C 04 B2 AC F6
SFN : 405.1	PUCCH - UCI HARQ ACK
	PUCCH Format 1 A n PUCCH = 16
SFN : 406.2	PUCCH - UCI SR
	N PUCCH RB = 2
SFN : 406.6	PDCCH - DCI Format 0
	PDCCH DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 406.7	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 406.8	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 406.9	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.0	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.0	PUSCH - RRC Connection Setup Complete (First Segment)
	MAC = 3A 3D 01 22 10 88 00 00 20 Sub Header 0

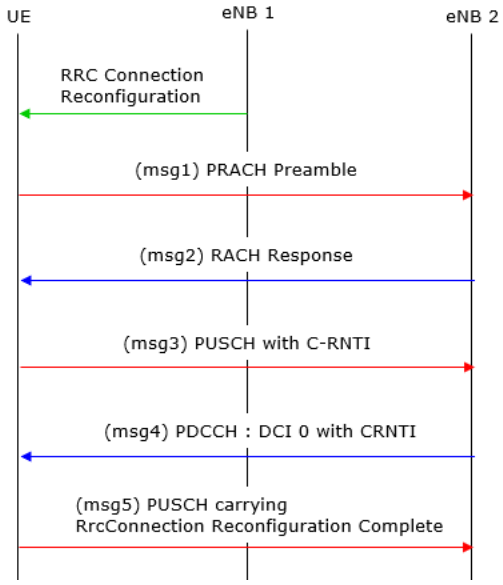
	R = OK E = 1 LCID = 26 (Power Headroom Report) Sub Header 1 R = OK E = 1 LCID = 29 (Short Buffer Status Report) Sub Header 2 R = OK E = 0 LCID = 1 (identity) Power Headroom R = OK Power Headroom --> 11 dB <= PH <= 12 dB Short Buffer Status Report LCG ID = 0 Buffer Size 16 --> 91 < BS <= 107 RLC AMD = 88 00 00 20 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 1 (First Byte of the Data Field corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 0 PDCP-CP-SRB = 00 20
SFN : 407.1	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 01 20 80 01 00 59 17 Sub Header 0 R = OK E = 0 LCID = 1 (identity) RLC AMD = 98 01 20 80 01 00 59 17 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 1 PDCP-CP-SRB = 20 80 01 00 59 17
SFN : 407.1	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.1	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 02 39 45 E5 34 0B 07 Sub Header 0 R = OK E = 0 LCID = 1 (identity) RLC AMD = 98 02 39 45 E5 34 0B 07

	D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 2 PDCP-CP-SRB = 39 45 E5 34 0B 07
SFN : 407.2	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.3	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 02 41 02 0B F6 03 02 Sub Header 0 R = OK E = 0 LCID = 1 (identity) RLC AMD = 98 03 41 02 0B F6 03 02 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 3 PDCP-CP-SRB = 41 02 0B F6 03 02
SFN : 407.3	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180540) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 1 (True) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.4	PHICH ACK
	....
SFN : 407.4	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0180440) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 96 MCS 2, RV 0 NDI = 0 (False) TPC = 1 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.4	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 04 27 80 01 00 D0 CC Sub Header 0 R = OK E = 0 LCID = 1 (identity) RLC AMD = 98 04 27 80 01 00 D0 CC D/C = 1 (Data PDU) RF = 0 (AMD PDU)

	P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 4 PDCP-CP-SRB = 27 80 01 00 D0 CC
SFN : 407.5	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 3D 01 0E 98 05 71 51 04 E0 Sub Header 0 R = OK E = 1 LCID = 29 (Short Buffer Status Report) Sub Header 1 R = OK E = 0 LCID = 1 (identity) Short Buffer Status Report LCG ID = 0 Buffer Size 14 --> 67 < BS <= 78 RLC AMD = 98 05 71 51 04 E0 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 5 PDCP-CP-SRB = 71 51 04 E0
SFN : 407.5	PHICH ACK
	.....
SFN : 407.5	PDCCH - DCI Format 0
	DCI Format 0 (Hex : 0246280) Format 0 Hopping Flag = 0 (False) RB Allocation of 1st Slot in UL subframe = 145 MCS 17, RV 0 NDI = 0 (False) TPC = 2 Cyclic Shift for DMRS = 0 CQI Requested = 0 (False)
SFN : 407.6	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 06 E0 C0 40 00 21 02 Sub Header 0 R = OK E = 0 LCID = 1 (Identity) RLC AMD = 98 06 E0 C0 40 00 21 02 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 6 PDCP-CP-SRB = E0 C0 40 00 21 02
SFN : 407.6	PHICH ACK
	....
SFN : 407.7	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 07 03 D0 11 D1 27 1A Sub Header 0 R = OK E = 0 LCID = 1 (Identity) RLC AMD = 98 07 03 D0 11 D1 27 1A

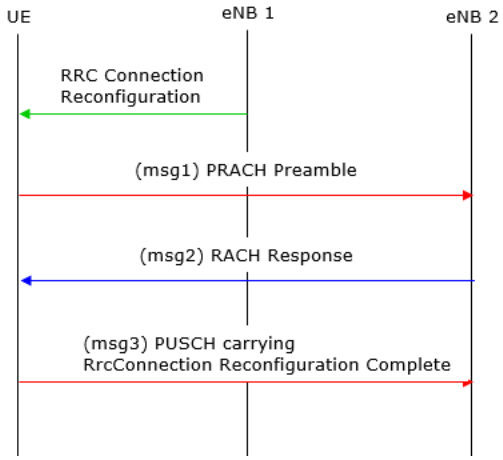
	D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 7 PDCP-CP-SRB = 03 D0 11 D1 27 1A
SFN : 407.7	PHICH ACK
	.....
SFN : 407.8	PHICH ACK
	.....
SFN : 407.8	PUSCH - RRC Connection Setup Complete (Mid Segment)
	MAC = 01 98 08 80 80 21 10 01 00 Sub Header 0 R = OK E = 0 LCID = 1 (Identity) RLC AMD = 98 08 80 80 21 10 01 00 D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 0 (Status Report Not Requested) Fl = 3 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field does not corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 8 PDCP-CP-SRB = 80 80 21 10 01 00
SFN : 407.9	PUSCH - RRC Connection Setup Complete (Last Segment)
	MAC = 3E 21 36 1F 00 00 00 B0 09 00 10 81 06 00 00 00 00 83 06 00 00 00 00 .... Sub Header 0 R = OK E = 1 LCID = 30 (Long Buffer Status Report) Sub Header 1 R = OK E = 1 LCID = 1 (identity) F = 0 (False) L = 54 Sub Header 2 R = OK E = 0 LCID = 31 (Padding) Long Buffer Status Report Buffer Size #0 = 0 (BS = 0) Buffer Size #1 = 0 (BS = 0) Buffer Size #2 = 0 (BS = 0) Buffer Size #3 = 0 (BS = 0) RLC AMD = B0 09 00 10 81 06 00 00 00 00 83 06 00 00 00 00 .... D/C = 1 (Data PDU) RF = 0 (AMD PDU) P = 1 (Status Report Requested) Fl = 2 (First Byte of the Data Field does not corresponds to the first byte of a RLC SDU. Last byte of Data field corresponds to the last byte of a RLC PDU) E = 0 (False) SN = 9 PDCP-CP-SRB = 00 10 81 06 00 00 00 00 83 06 00 00 00 00 ....

< RACH Procedure on Handover - Contention Based >



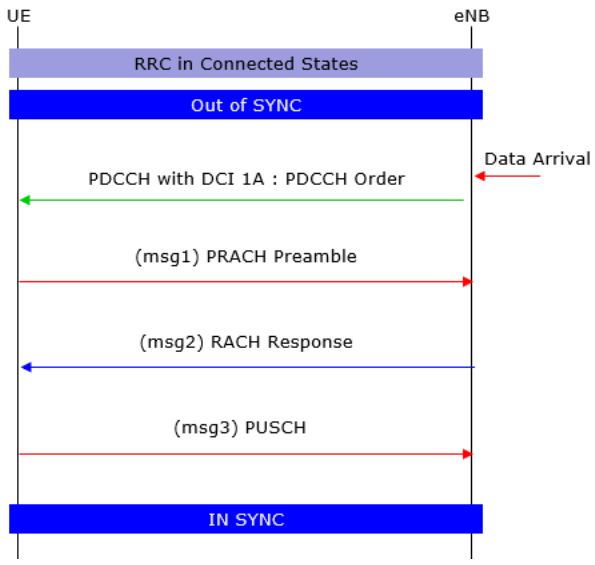
< RACH Procedure on Handover - Contention Free >

For the complete handover procedure with this type of RACH proceder, refer to this note. In most handover, this kind of contention free RACH procedure is used since UE and eNB is already in sync with C-RNTI.

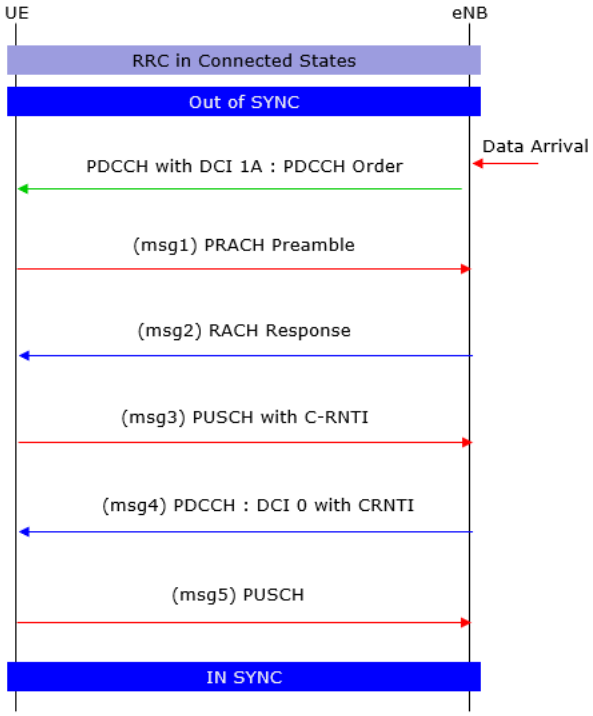


<RACH Procedure on DL Data Arrival when Out-of-Sync - Contention Free >

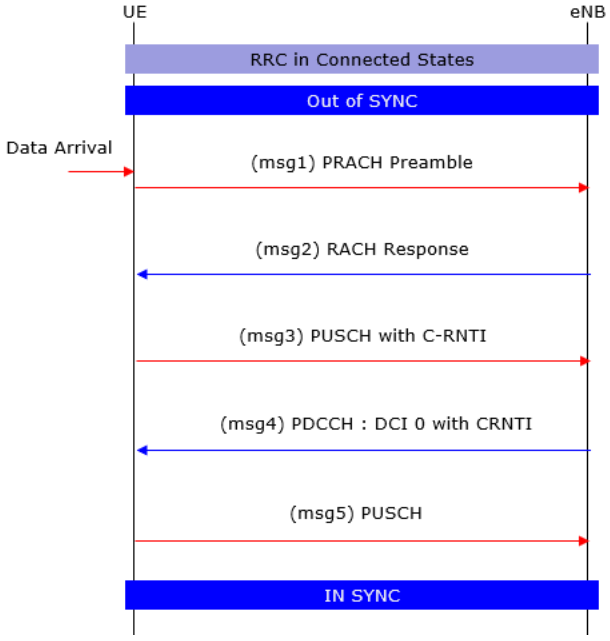
PDCCH order is a mechanism for eNB to trigger RACH procedure in connected mode in order to recover the sync. Refer to PDCCH Order page for the details of RACH Triggered by PDCCH Order. I wrote a separate page because there are pretty much details on PDCCH Order itself.



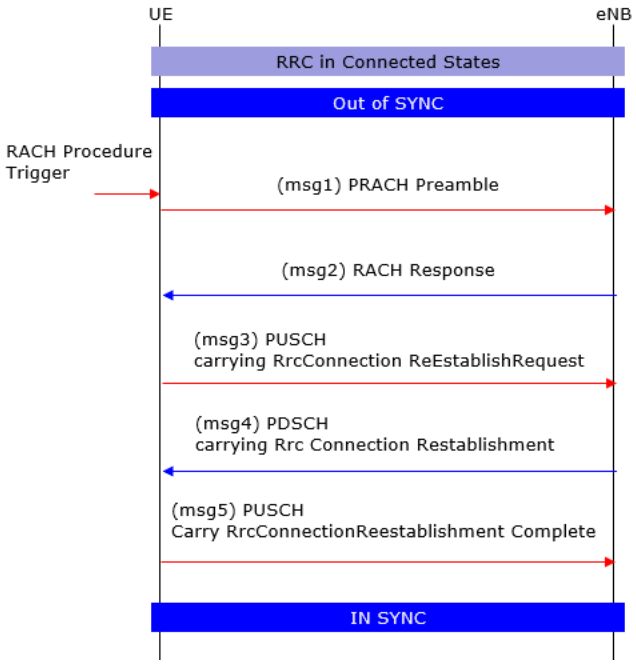
<RACH Procedure on DL Data Arrival when Out-of-Sync - Contention Based >



<RACH Procedure on UL Data Arrival when Out-of-Sync >



<RACH Procedure on RRC Connection Re-establishment when Out-of-Sync >

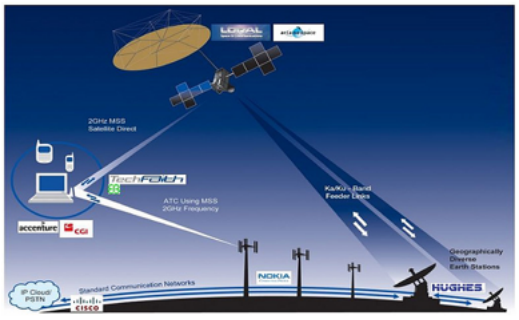


PRACH Optimization in Cell Planning

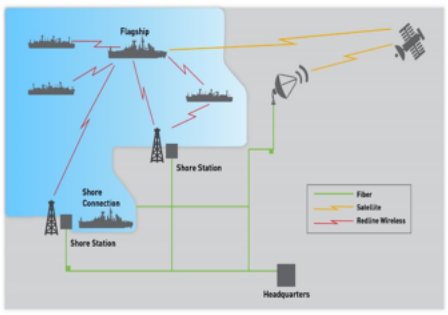
This section would be the one you can apply all the knowledge of PRACH parameters you have learned. As any other theories in the text book, it is not always easy to digest and apply those parameters in real life cell design and parameter optimization. To be honest, I don't have any personal experience of applying those parameters in real life cell design and parameter optimization. Especially for this section, I got valuable advise from an expert (Jared Cho) and used his comments as a skelleton and I will keep adding as I learn more. Also, I am waiting for the comments/advise from the readers who has experiences/experties in this subject.
My approach is to set a few extreme situations and identify what would be the dominant factors (parameters) in those situation. Following is several extreme cases that I could think of as of now (probably add more). First think of what kind of RACH parameters will be the dominant factors for each of these cases on your own.

< Cell Radius and Multi Path condition for different cell sites >

< A > Extremely Large Cell Radius



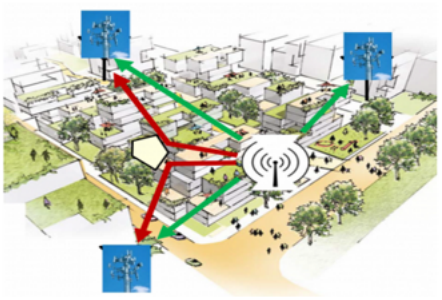
< B > Extremely Large Cell Radius



< C > Extremely High Doppler

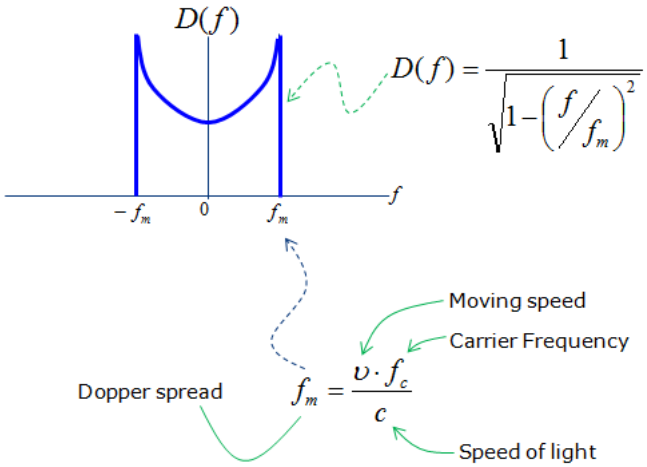


< D > Large Fading/Multipath



In Case < A> and < B >, the cell redius would be extremly large from a few tens of km even up to 100 km in extrem case. Even though the cell radius is very large in this case, the path between the transmitter and reciever tend to be line of sight. So, in this case I think Preamble format can be a dominant factors in terms of covering wide range and maximun TA (Timing Advanced) value in RAR can be the domant factor as well. Due to the long distance between UE and eNB, there tend to be wide range of timing delay. So TA range that can compensate those wide range of timing variation. In addition, to compensate the possible delay spread due to long traveling distance, N_CS (zeroCorrelationZoneConfig) would play an important role as well. The larger cell radius is, you may pick higher values for zeroCorrelationZoneConfig. For example, I was told that a cell can cover around 100 km with preamble format 1 and zeroCorrelationZoneConfig = 15

In Case < C > which is the case where High Speed Training model can apply. In this case, both UE and eNB can experience a large doppler shift (expecially when the train just passes eNB). In this case, turning on 'HighSpeedFlag' (i.e, HighSpeedFlag = True) would be important and selecting proper N_CS value depending on situation would be important. Since the doppler shift is a determining factors of parameter optimization, you may easily understand that not only the speed of the motion but also the carrier frequency will play the important role. As you know, Doppler shift is determined by both moving speed and Carrier Frequency as shown below.



Then you may ask exactly 'Is there any clear cut borderline between HighSpeedFlag=True and HighSpeedFlag=False in terms of Doppler Spread value ?'. I haven't found any document clearly defining on this. However, once you specify a certain doppler shift value as a borderline and you have a specific carrier frequency, you can easily figure out the speed limit for enabling/disabling the HighSpeedFlag. For example, if you set the doppler spread of 200 Hz as the border line, you can determine the speed limit as follows.

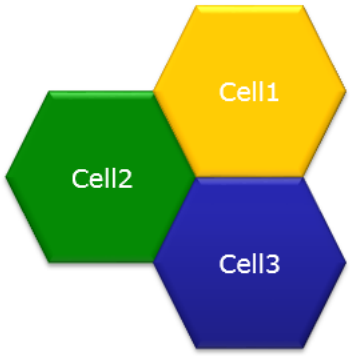
Carrier Frequency (Ghz)	Moving Speed (km)
0.85	254
1.8	120
2.1	103
2.6	83

Now let's see an example showing the configuration of other PRACH parameters which is applicable in live network cell planning. I don't have personall experience with live network cell planning and Jared Cho happily contributed his experties here as well.

< Example of RACH Parameters of neighbouring cells >

Parameter	Value
PRACH Config Index	3
Root Sequence Index	10
ZeroCorrelationZoneConfig	12
HighSpeedFlag	FALSE
Prach-FreqOffset	4

Parameter	Value
PRACH Config Index	3
Root Sequence Index	30
ZeroCorrelationZoneConfig	12
HighSpeedFlag	FALSE
Prach-FreqOffset	4



Parameter	Value
PRACH Config Index	3
Root Sequence Index	20
ZeroCorrelationZoneConfig	12
HighSpeedFlag	FALSE
Prach-FreqOffset	4

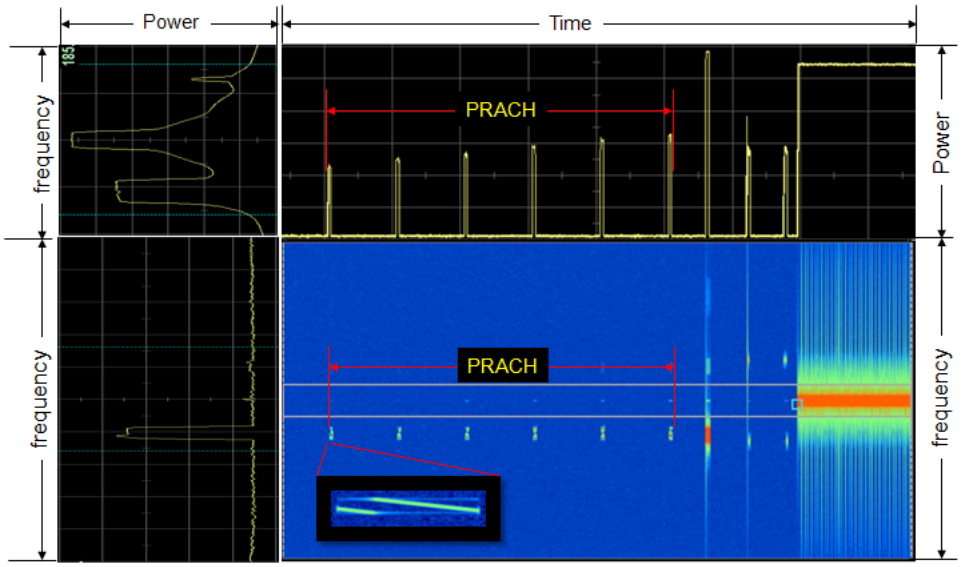
Let's take a look at some of the configurations set in this example. (In this example, it is assumed that the preamble format is set to be Format 0)
First you would notice that all the cells has same PRACH config Index. Are these required to be same ? No, you configure them differently depending on situation. If you set them differently (especially in such a way that each of the

cells receive PRACH at different subframes), it would be helpful to reduce any possible interference caused by PRACH for other cell. However, in this case a PRACH in one cell can cause interference to PUSCH in other cells.
Second ZeroCorrelationZoneConfig is set to be same in this case. These are not required to be the same either. In this example, we assume that the radius of every cell is almost same. Considering that Preamble Format 0 has roughly 14 km, ZeroCorrelationZoneConfig is set to be 12 to match the coverage and Root Sequence Index for each cell is set with a certain interval (the interval of 10 in this case).
Lastly prach-FrequencyOffset is set to be 4. This can be set as any possible value, but it is recommended that PRACH would not cause any interference with the location of PUCCH.

PRACH RF Snapshot

To capture this spectrum, I configured the setup where UE failed to receive RAR from network for several times. As you see here, UE keep transmitting PRACH with the increment of power specified by powerRampingStep in SIB2 until it successfully detects RAR. It will keep retransmitting until the number of transmission hits preambleTransMax. The interval between a preamble and next preamble is determined by backoff indicator.

< PRACH on Spectrum Analyzer >



RACH in srsRAN

If you are interested in real implementation of RACH procedure at source code level, following API in srsRAN can be a good reference.

- prach_Tcp[5] -> \lib\src\phy\phch\prach_tables.h
- prach_Tseq[5] -> \lib\src\phy\phch\prach_tables.h
- prach_Ncs_unrestricted[16] -> \lib\src\phy\phch\prach_tables.h
- prach_Ncs_restricted[15] -> \lib\src\phy\phch\prach_tables.h
- prach_Ncs_format4[7] -> \lib\src\phy\phch\prach_tables.h
- prach_zc_roots[838] -> \lib\src\phy\phch\prach_tables.h
- prach_zc_roots_format4[138] -> \lib\src\phy\phch\prach_tables.h
- srsran_prach_gen_seqs() -> \lib\src\phy\phch\prach.c
- srsran_prach_gen() -> \lib\src\phy\phch\prach.c
- generate_buffer() -> \srsue\src\phy\prach.cc
- prepare_to_send() -> \srsue\src\phy\prach.cc
- is_ready_to_send() -> \srsue\src\phy\prach.cc
- prach_send() -> \srsue\src\phy\phy.cc
- resource_selection() -> \srsue\src\stack\mac\proc_ra.cc
- preamble_transmission() -> \srsue\src\stack\mac\proc_ra.cc
- tb_decoded_ok() -> \srsue\src\stack\mac\proc_ra.cc
- state_response_reception() -> \srsue\src\stack\mac\proc_ra.cc
- process_timeadv_cmd() -> \srsue\src\stack\mac\proc_ra.cc

3GPP Standard for RACH Process

3GTS 36.300 (10.1.5) : Overall description of RACH Process. Read this first.
3GTS 36.211 (5.7) : RRC Messages and IE (Information Elements) which are involved in RACH process.
3GTS 36.213 (6) : MAC Layer Procedure related to RACH Process.