Cellular IoT Technology

Cellular Network Basics

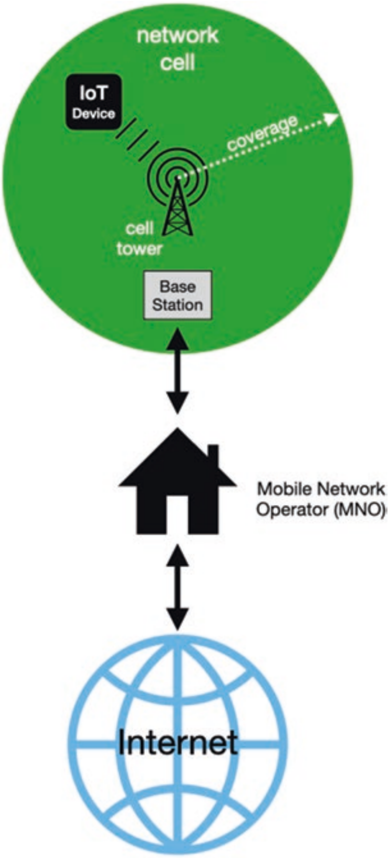
A cellular network is made up of a large number of signal areas called cells. These cells join or overlap each other. Within each cell you will find a base station or **cell tower** which sends and receives data. A mobile phone or IoT device usually connects to the closest available **base station**, if not congested by other user devices. Base stations are gateways for wireless data to the MNO network. For IoT devices, a connected base station is acting as the entry point for Internet communication (Fig. 1).

Cells are owned and managed by MNOs which are usually private companies. Acquired license provides exclusive usage rights for a certain carrier frequency range to the MNO. Connectivity services offered by an MNO will be usage-based, i.e., users will have to agree on commercial conditions and sign a corresponding contract with selected MNO, i.e., a **subscription plan** applies. Purchase of a national cellular license might include some governmental obligations, but normally each MNOs will take **network coverage** decisions based on business perspective, i.e., based on customer demand. With other words: an MNO will not extend network coverage into areas where nobody will use it. In addition, each cell has a limited **capacity**, i.e., it provides only a limited number of **channels** for simultaneous user connections. As a result, each MNO will create a cell floorplan and manage cell deployment accordingly—in an effort to offer good network coverage and service for all subscribers.

A cell tower is a platform where antennas and other hardware are being mounted. Physically, this could be a dedicated mast or a building. In general, cell range (or cell size) depends on used antenna and applied output power (which might be restricted/limited by regional law). In rural areas, a single network cell will be able to cover an area of up to 35 km, but density of cells will be low. With help of a range extender, Australian MNO Telstra even claims to be the “first operator globally to implement a 120 km radius on NB-IoT” [3]. By nature, cell coverage area will be small in urban areas because high-frequency radio waves cannot easily pass

K. Heins, *NB-IoT Use Cases and Devices*, https://doi.org/10.1007/978-3-030-84973-3_2

Fig. 1 Network cell to Internet



building walls easily. On the other hand, high population in urban areas means a lot of users might request service at the same time. This means that a new connection request would be rejected by a cell if its capacity is exhausted. This gap will be compensated by additional cells within reach of a user terminal, i.e., it might not connect to the closest cell. Instead, it will allocate the next available channel offered by another cell within reach. This way, a high density of small cells in urban areas will enable massive connectivity (Fig. 2).

In general, device **power consumption** for uplink data transmission correlates with distance to cell tower. Closer distance resp. a small cell means that less output power is required by a user device to send data. At least for battery-powered stationary IoT devices, location resp. distance to attached cell tower might have a significant impact on battery lifetime. In urban areas, so-called femtocells (10 m range) or picocells (200 m range) are used to cover buildings which are also supporting IoT deployments.

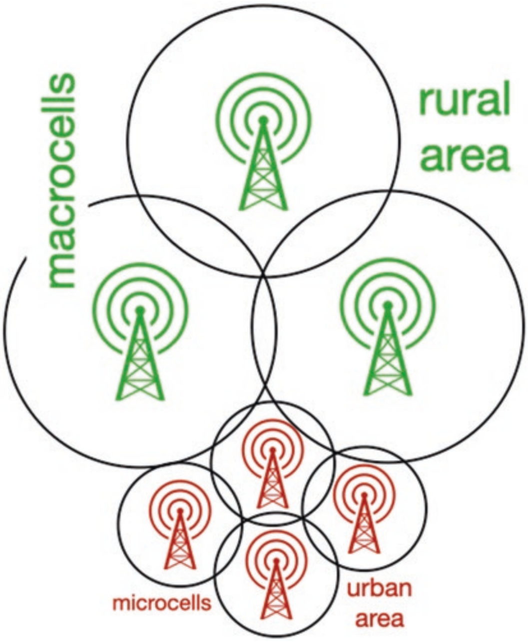


Fig. 2 Cellular network grid

Modem Interface

Before we start to look closer at cellular network technology, we have to travel back in time for a minute and recall the original meaning of some telecommunications terms and mechanisms. For example, in technical documentations of NB-IoT cellular networks modules still terms like DTE (Data Terminal Equipment) or MT (Mobile Terminal) are being used although not quite applicable to M2M applications of today. Another example are so-called AT commands which been invented back in 1981 to control Hayes modems to transmit data via phone line at a data rate of 300 baud. At that time, AT control commands have been used by a computer terminal for dialing and hanging up. Modem connection was a 9-pin RS-232 serial interface using ±12 V levels which is not common practice any more, but UARTs and RS-232 signals (TxD, RxD, CTS, RTS) are still being used by modern IoT cellular network modules. Figure 3 is explaining both use scenarios and some common terms and acronyms.

Originally, data transfers always have been initiated and controlled by a human sitting in front of a computer or terminal, and user information was entered or prepared for transmission. This communication scenario still exists, esp. with mobile terminals, i.e., smart phones. This is why an end device is sometimes called MT (mobile terminal) or DTE, even if—in case of machine-to-machine communication (M2M)—an IoT application takes over and “replaces” the human user. In fact, the **IoT application is an embedded software program** running inside the IoT device

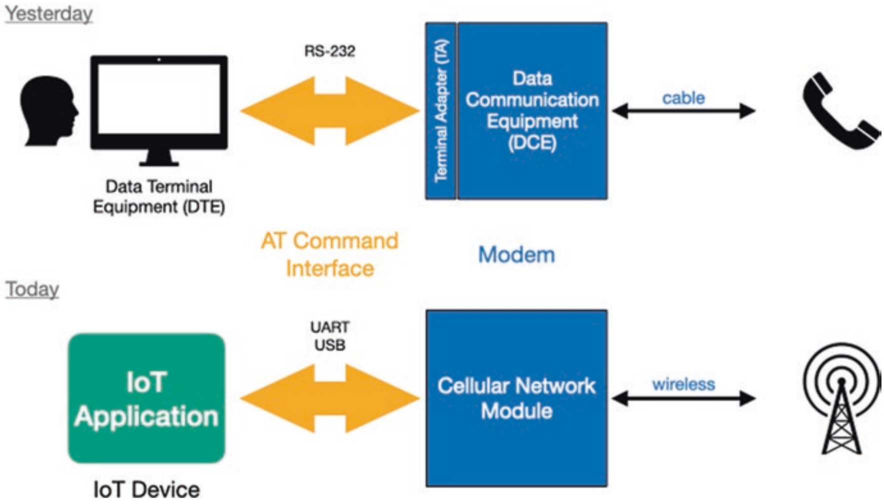


Fig. 3 AT Command Interface—yesterday and today

(see chapter "Ingredients for NB-IoT Design Concepts"). Term "modem" is still valid because this component is still used to convert digital data for telecommunication, but transmission medium telephone line has been replaced by wireless cellular network. Term "Terminal Adapter (TA)" was used for the connection part of the dial-up modem in the past, and still sometimes used as a misleading name for the **AT command interface** function of a cellular network modem.

AT commands are still used as a standardized control API for cellular network interface subsystems, which are key components for IoT devices incl. NB-IoT (see section "NB-IoT Cellular Network Modules" of chapter "Ingredients for NB-IoT Design Concepts"). AT-based communication is using UART or USB serial interfaces inside the IoT device, mastered by a dedicated application MCU.

Standard AT commands are used to control modem functions and network services. They are specified by 3GPP TS 27.007 "AT command set for User Equipment (UE)." Initial release was published back in 1999. Further evolution of the 3GPP standard and additional AT commands is reflected by newer releases and further specifications. Consequently, TS 27.007 is being updated regularly. From an application point of view, all standardized AT commands should be available and corresponding functions should have been implemented by every commercial cellular modem. In addition, related network services should have been implemented by the network infrastructure as well, otherwise functions requiring cooperation with network (esp. NB-IoT power saving mechanisms) will not work as expected.

On top of standardized functions, manufacturers of IoT cellular network modules can add extra AT commands to control proprietary functions or to utilize complementary software functions. In fact, a **cellular network module** today is a highly integrated, mixed-signal MCU-based subsystem which is—besides other functional elements—also including the modem function. Operating software is field-updatable and sometimes even customizable, i.e., open for application software. Consequently, most manufacturers are offering cellular modem extensions like Internet software stacks and communication protocols. In practice, the IoT device can obtain a virtual IP socket from the network module which is acting as a TCP/IP endpoint on behalf of the device. Or, for example, secure end-to-end communication can be handled by the module via embedded SSL/TLS stack.

From an IoT application point of view, this is added value because customer development can focus on application expertise rather than standard connectivity ingredients. This kind of software building blocks are bundled with most IoT cellular network modules, and they are supported by a proprietary set of AT commands. Also, IoT protocols like MQTT or FTP function for data transfer or firmware update are not standard and handled differently by manufacturers.

For NB-IoT device designers, selecting a powerful network module with a comprehensive AT command set is a key enabler for fast turnaround and time-to-market.

During field operation, mentioned embedded IoT application will submit AT commands to the modem via UART system call. But during development or for evaluation or test purposes, users can also issue AT commands via a hyper terminal using a command-line interface, e.g., on a Windows or Mac PC.

Examples

1. Check if device is connected to a network and registered:

Command:

```
AT+COPS?
```

Response:

```
+COPS: 0,0,"vodafone IT",7
OK
```

2. Change operating LTE band (for next registration attempts):

```
AT+UBANDSEL=1800,2100,2600
```

Response:

```
OK
```

A dedicated section “AT Command Interface” of chapter “Ingredients for NB-IoT Design Concepts” will take a comprehensive look at AT command syntax and provide more examples.

NB-IoT Technology

Until early 2015, GSM/GPRS/EDGE had been the main cellular technology of choice for serving wide-area IoT use cases. At this time, GPRS was a mature technology with low modem cost. But market demand for low-power wide-area networks (LPWANs) was increasing and GPRS was challenged by alternate technologies in unlicensed spectrum like Sigfox, LoRa. Anticipating this new competition, 3GPP started a feasibility on “Cellular system support for ultra-low complexity and low throughput Internet of Things” [4].

These efforts resulted in ambitious objectives for cellular IoT features in a post-GPRS era. Main goals were as follows (Fig. 4):

1. Improve cell **coverage** for rural areas and **penetration** of buildings down to ground level.
2. Reduce **device complexity and cost** in order to enable massive IoT applications.
3. Network **latency** should not exceed 10 s.
4. Achieve high **capacity** for a massive number of IoT devices generating a small amount of data.
5. Minimize device **power consumption** to achieve 10 years lifetime of a 5 Ah battery.
6. **Reuse existing LTE network** infrastructure and upgrade with new IoT features by software only.

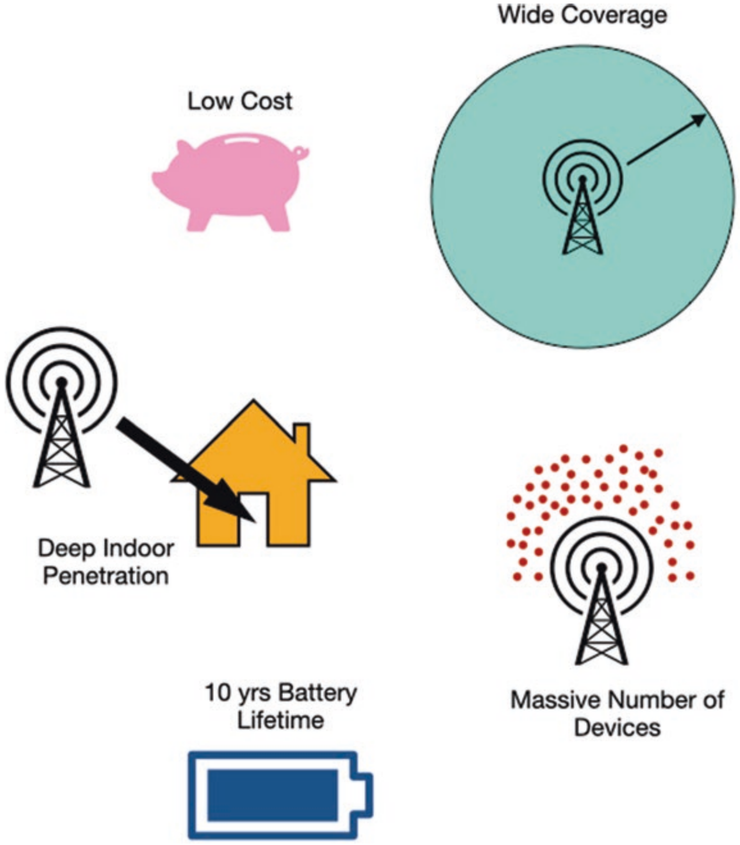


Fig. 4 NB-IoT objectives

As a result of related standardization work and worldwide agreement, **NB-IoT** (= "Narrowband IoT") has been defined as an add-on to 4G/LTE in 3GPP Release 13.

Network Deployment Options

For network operators, NB-IoT has been designed with various deployment options in mind. In a stand-alone scenario, an NB-IoT carrier would replace a GSM carrier. With a carrier bandwidth of just 200 kHz (same as for GPRS), an NB-IoT carrier can also be deployed within an LTE carrier or in an LTE guard-band. Available options are illustrated in Fig. 5.

Stand-alone deployment in a GSM band is an option for operators running GSM and LTE networks in parallel. In this case, one or more GSM carriers can be "refarmed" to the LTE network so that they can be used to carry NB-IoT traffic.

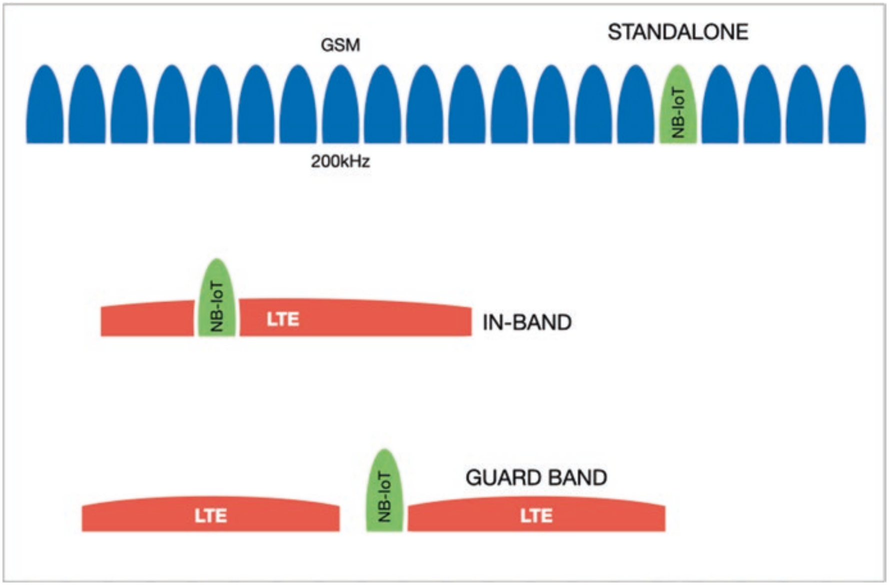


Fig. 5 Spectrum usage deployment options

Later, when time is right, the MNO can refarm his complete GSM spectrum for use by LTE. According to Ericsson [1], a leading manufacturer of cellular network equipment, this migration is a seamless process even with deployed NB-IoT carriers in the GSM spectrum as it will not impact NB-IoT devices at all and will continue to work in the LTE spectrum afterwards.

In general, for LTE operators the **in-band** option probably is best choice because it allows flexible assignment between LTE and NB-IoT. So, for example, an operator can start with a stand-alone deployment of its NB-IoT service in a GSM band and subsequently migrate these NB-IoT bands to an in-band LTE deployment depending on increasing customer demand.

A third alternative is to focus deployment of NB-IoT in LTE **guard-bands** which are adjacent to each LTE carrier. In fact, physical NB-IoT layers have been designed to coexist in LTE guard-bands without causing interference. Guard-band deployment can be done without affecting the capacity of the LTE carrier.

All three deployment scenarios are transparent to non-NB-IoT devices. Consequently, LTE devices that do not implement NB-IoT functionality simply do not see the NB-IoT channel inside the main LTE bandwidth or in the guard-band. At the same time, legacy GSM devices will not see an NB-IoT carrier if used alongside 180 kHz GSM carriers. Such devices will only see noise where NB-IoT is active [5].

By the way, with 3GPP Rel. 15 the coexistence of 5G with NB-IoT has been secured. It provides the opportunity to introduce 5G in carriers where NB-IoT is already in operation. In future, existing NB-IoT deployments can continue to provide IoT connectivity in 5G networks as a low-end and low-complexity option [6].

Cell Capacity

NB-IoT has been designed to support massive deployment of IoT devices, e.g., for Smart City applications, but also for big events where lots of people gather in small areas (with wearable IoT devices). The capacity requirements target in 3GPP TR43820 [4] for Release 13 has been set to 40 devices per household which corresponds to 52,500 devices per cell or 60,680 devices per km^2. This requirement assumes that base stations are located on a hexagonal cell grid with an intersite distance of 1732 m resulting in a cell size of 0.87m^2 (Fig. 6).

How can this be achieved? The idea is to multiplex traffic of lots of devices altogether in an efficient manner. As a result, NB-IoT supports a range of data rates which depends on channel quality (signal-to-noise-ratio) and allocated bandwidth.

For the **downlink** (short: “DL”), at physical layer, NB-IoT fully inherits LTE transmission scheme. QFDM multiplexing is applied using a 15 kHz subcarrier spacing with a normal cyclic prefix CP. As a result, each of the QFDM symbols consists of 12 subcarriers occupying the bandwidth of 180 kHz. Seven QFDMA symbols are bundled into one 180 kHz slot, resulting in a resource grid illustrated in Fig. 7. All CIoT devices share the same power budget and may simultaneously receive base station transmissions.

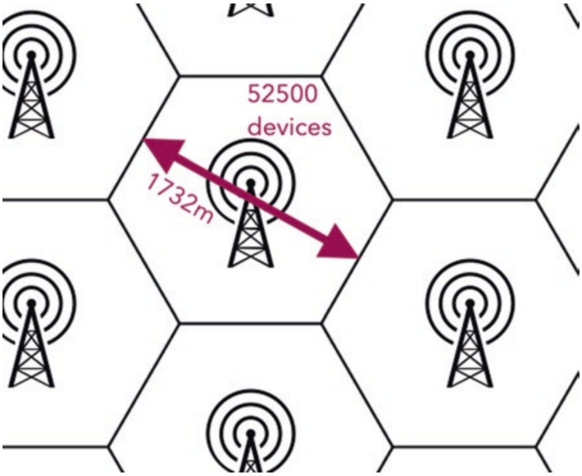


Fig. 6 Cell capacity of 60,680 devices per km^2

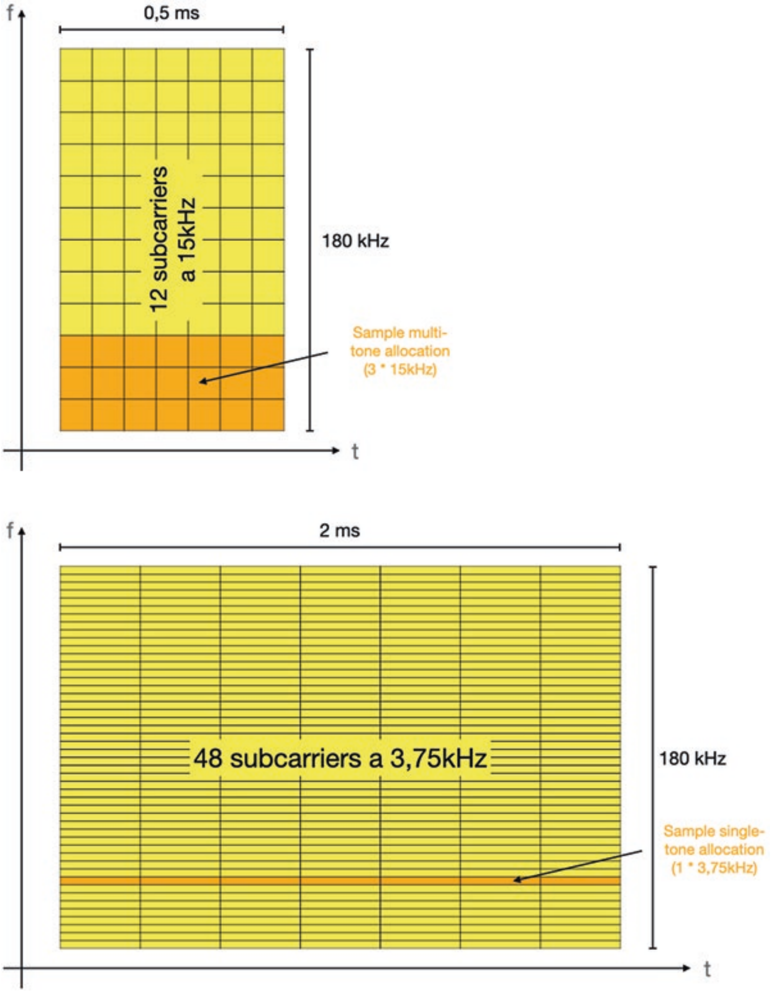


Fig. 7 Resource grids for 15 and 3.75 kHz subspacing

In the **uplink** (short: “UL”), however, each device has its own power budget which can be combined by multiplexing traffic of several devices in same cell. These devices can concatenate their transmission energy to a narrower bandwidth [1]. The idea is to allocate only small amounts of bandwidth to specific devices in order to increase overall capacity without performance degradation. Practically, instead of resource blocks with an effective bandwidth of 180 kHz, NB-IoT allocates subcarriers (or “tones”).

In the UL, SC-FDMA mechanism is applied, either with a 3.75 kHz or 15 kHz subcarrier spacing. The 3.75 kHz option has been added exclusively for NB-IoT, it increases **flexibility of bandwidth allocation** even further. The LTE eNodeB decides which one to use [7]. For 15 kHz subcarrier spacing the UL resource grid is identical as for DL, but for 3.75 kHz it has a different structure:

There are still 7 OFDM symbols within a slot. But, according to the OFDM principles, the symbol duration for 3.75 kHz subcarrier spacing has four times the duration compared to 15 kHz, which results in a slot length of 2 ms.

Each NB-IoT device can be scheduled on one subcarrier (“single-tone”) or more subcarriers (“multi-tone”) in the uplink. Smallest amount of bandwidth to allocate to a device is a subcarrier of 3.75 kHz. Allocation of multiple subcarriers works only with 15 kHz ($n \times 15$ kHz, $n = [3, 6, 12]$) and will increase transmission data rates of selected devices accordingly. This option is adding extra flexibility in cellular environments handling devices both in good and in bad coverage areas. Devices enjoying good coverage and configured for multi-tone will be able to finish data transfer 12 times more quickly (vs. single-tone) and release their allocation to others in a shorter period of time. This approach will add capacity in certain macrocell scenarios, but will not be required in areas with a dense cell grid or other good coverage scenarios, e.g., where all devices are located within the standard LTE cell range.

According to simulations performed by network equipment manufacturer Ericsson [1], evaluations have shown that a “standard deployment” can support a **density of 200,000 NB-IoT devices within a cell**, i.e., achieved capacity is exceeding original goal by far (four times higher). Observations in real-world mass deployments will provide further insights, but for the time being it seems that NB-IoT multiplexing scheme in combination with adjustable data rates will meet capacity goals.

Coverage Extension

Another important NB-IoT objective is to achieve *adequate* coverage at a maximum coupling loss (MCL) of 164 dB which represents a **20 dB coverage enhancement** compared to GSM/GPRS. Practically, this will result in wider data transmission range as well as deeper penetration into buildings, underground assets, tunnels, etc. In fact, 20 dB improvement corresponds to a **sevenfold increase in coverage area for an open environment**, or roughly the loss that occurs when a signal penetrates the **outer wall of a building**.

The reduction of NB-IoT bandwidth already helps a lot because the LTE eNodeB maintains the same transmission power as in LTE case (43 dBm). In fact, it increases the power density ratio (PSD). In uplink, NB-IoT bandwidth can go down to 3.75 kHz, GPRS is at 200 kHz. Consequently, PSD ratio is 5.3 which is around 7 dB which is a first contribution to coverage enhancement goal of 20 dB.

Second contribution is **repeated transmission of data packets** depending on individual radio conditions, i.e., NB-IoT devices in difficult areas will retransmit IoT message several times in order to increase eNodeB chance to decode the received message correctly. For this purpose, three **coverage enhancement (CE) levels** have been specified: CE0, CE1, and CE2. CE level 0 represents the standard LTE coverage level, and CE level 2 corresponds to the worst case, where the coverage may be assumed to be very poor.

Note: It is up to the network how many CE levels are defined. A list of power thresholds for the received reference signals is broadcasted in the cell for each CE level.

The main impact of the different CE levels is that the messages have to be repeated several times. In case of CE0, since we are in standard LTE conditions, the device will not have to repeat the data. In CE1 the device would be out of LTE coverage, then it needs to repeat the packets 2, 4, 8, or 16 times; finally, in CE2, we have the worst channel conditions, so the device must repeat data 16, 32, or 128 times (Fig. 8).

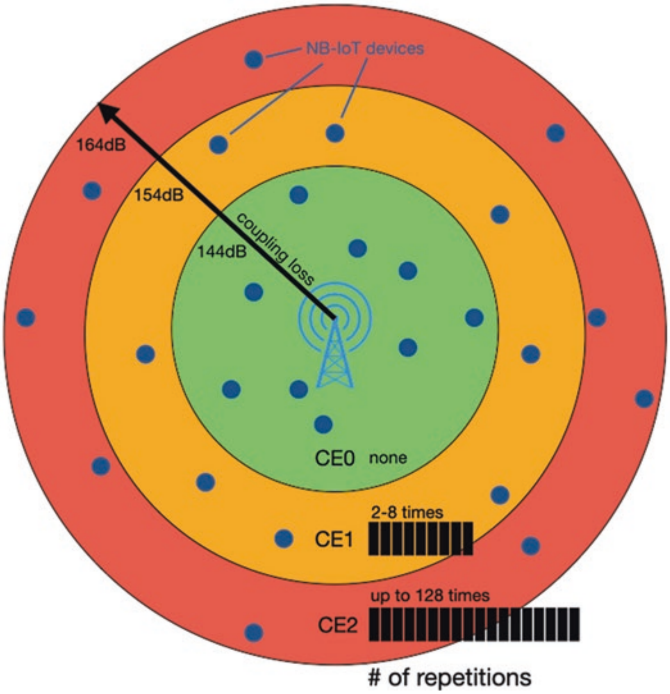


Fig. 8 Coverage enhancement levels (CE)

The repetition mechanism used by NB-IoT is a **hybrid automatic retransmission request (HARQ)** scheme. HARQ is a combination of error detection (ED), forward error correction (FEC), and automatic repeat request (ARQ) error-control. ED and/or FEC information will be added to each IoT message prior to transmission in order to detect and correct an expected subset of all errors that may occur, while the ARQ method will lead to retransmissions as a fallback in case of uncorrectable errors in original message. In HARQ type 1, an erroneous data frame will be rejected and will be re-transmitted until received error-free or fully corrected or when the maximum number of retransmissions is reached. For NB-IoT uplink transmission, a more sophisticated version is being used: HARQ type 2. In fact, HARQ 2 is **alternating embedded ED and FEC information** in re-transmitted data frame case-by-case. FEC bits are only transmitted on subsequent retransmissions as needed, i.e., re-transmitted will be shorter for devices facing good signal conditions. In this case, missing FEC info will reduce cell capacity required for coverage enhancement.

A repetition factor of two results in less than 3 dB. Thus 128 repetitions are around 13 dB [8]. So, adding these 13 dB to mentioned 7 dB gain due to increased PSD ratio, NB-IoT achieves target of 20 dB coverage enhancement vs. GSM/GPRS.

Network Selection

By nature, operation of a CIoT device requires connection to a network cell. At least in urban areas, not just one but multiple cellular networks might be available, so which one to select? Selection criteria are different and closely related to network coverage requirements of each IoT application, esp. where devices will be located and if they are mobile or not. These considerations will lead to selection of an appropriate network partner (MNO or MVNO, see also section “Roaming and MVNO” of chapter “NB-IoT Network Deployment”) to ensure that **all** IoT locations are supplied with sufficient network coverage (i.e., signal strength and quality). Consequently, an associated subscription plan will be already in place whenever an CIoT device will be powered on for the first time and looks for a cell to connect to. In fact, selected network partner will provide a SIM (SIM = Subscriber Identification Module) for CIoT device which is inserted during production and usually remains inserted during complete product life (see further information in extra section “SIM Card or Embedded eSIM” of chapter “Ingredients for NB-IoT Design Concepts”). Each SIM also contains a unique number IMSI (international mobile subscriber identity) which is used by the network to identify the user of the IoT device, i.e., the owner.

For cell selection, the device will first refer to several PLMN entries stored in SIM. PLMN stands for (“Public Land Mobile Network”) and allows to differentiate cellular networks offered by different MNOs using a unique PLMN code. A PLMN code is not just for operators *owning* a network infrastructure incl. cell towers, etc. like Swisscom (PLMN is “228 01”). Instead, also “virtual” MNOs (aka MVNOs) have their own PLMN identifier. So, for example, the PLMN code for specialized IoT-MVNO company Emnify is “295 09.”

Usually, after power up or after relocation from an uncovered area, a device will try to **connect again to the saved PLMN** it was previously registered on. If this PLMN is not available or connection attempt fails, another PLMN can be selected, either automatically or manually, depending on the PLMN priority information stored in SIM and/or signal strength. If automatic network selection mode is disabled, the user (resp. IoT application) can select from a list of choices presented by the modem.

In general, many aspects of cell scanning and registration procedure are user configurable and can be tuned via user API offered by cellular network modules (see section "Modem Interface") according to application needs. A stationary NB-IoT device, for example, might boot only once (during installation after deployment) and will remain camped on a specific cell during its complete product lifetime. In this case, also roaming can be disabled by default.

In order to support selection process, essential network system information (SI) is broadcasted by each LTE eNodeB to all user devices within reach. It contains a **Master Information Block (MIB)** and a couple of **System Information Blocks (SIBs)**, here is a short description of first blocks to be considered:

System Information Block	Content
MIB-NB	Operation mode info (stand-alone, in-band, guard-band), etc., information about SIB1-NB scheduling
SIB1-NB	Cell-access related information such as **PLMN**, **tracking area and cell identities**, access barring status, **thresholds for evaluating cell suitability**, etc., scheduling information regarding other SIBs
SIB2-NB	**Radio resource configuration (RRC) information** (incl. RACH-related configuration) for all physical channels that is common for all devices
SIB3-NB	Parameters required for intra-frequency, inter-frequency and I-RAT cell reselections
SIB4-NB	Information regarding INTRA-frequency neighboring cells (E-UTRA)
SIB5-NB	Information regarding INTER-frequency neighboring cells (E-UTRA)

In general, a user device should store a valid version of MIB-NB, SIB1-NB, and SIB2-NB through SIB5-NB. The other ones have to be valid if their functionality is required for operation. For instance, if access barring is indicated in MIB-NB, the user device needs to have a valid SIB14-NB.

Access Barring (AB) is an access control mechanism adopted in NB-IoT. It allows a PLMN to restrict, allow, or delay access according to assigned access class of a device (which is stored in SIM). For LTE, 15 access classes have been specified, 10 "normal" classes and 5 "special" classes. Class 12 is for "Security Services" and Class 13, for example, is for "Public Utilities" and might be used for Smart City IoT applications. An AB flag is provided in MIB-NB. If it is set false, all devices are allowed to access. If the AB flag is set true, the device must read the SIB-14 before it attempts to access the network, which provides specific barring information for access classes. The IoT device needs to check whether its access class is allowed to connect to the network. If not, the device should try again at a later point of time or select a different PLMN, if available.

As mentioned in section “Coverage Extension,” a device in bad coverage locations requires a significant number of repetitions to be configured for its upload traffic. During high network loads a PLMN might bar such devices and use available resources to serve more devices in good coverage locations instead. 3GPP Release 15 introduced a **coverage-level-specific barring** to exclude devices with a specific CE level (or worse). For this purpose, an RSRP threshold value is provided in SIB14-NB. If a device has measured RSRP below this threshold, it will be barred from this network.

As mentioned, applied subscription plan for an IoT project will determine which PLMNs should be used, i.e., usage cost will have significant impact on cell selection procedure. But if no priority PLMN is available or manual selection has been configured, the NB-IoT device will search for cells on appropriate frequency bands, reads the associated SIB information, selects a cell and measures the quality (i.e., noise) and the power level of the received NRS (NB Reference Signal). Relevant parameters for signal strength and signal quality are **RSRP (Reference Signal Received Power)** and RSRQ (Reference Signal Received Quality). RSRP is the measured power of the LTE reference signals spread across the broadband and narrowband portions of the spectrum. RSRP values, presented in dBm, are always negative, and the higher the number, i.e., the closer to zero it is, the higher the power of the signal. RSRP is a variation of parameter RSSI (Received Signal Strength Indicator) which is less relevant because it can be calculated with RSSI = RSPR/RSRQ.

In order to optimize cell selection procedure, local measurement of the actual signal strength at point of deployment should be part of the installation procedure for each IoT device. Then, measured RSRP value is compared to cell-specific threshold provided by SIB2-NB. For example, SIB2 might contain the following line:

```
pdsch-Config =
    referenceSignalPower : 15 dBm
```

In this case, the cell reference signal has been transmitted by eNB base station with 15 dBm output power. Let us assume that measured RSRP value at device target location is “−80 dBm.” Based on these numbers you can calculate path loss between cell and device: Path Loss = (**referenceSignalPower**) − (RSRP measured by user device) = 15 − (−80) = 95 dBm. In this case, the device can consider itself to be in good range (see Fig. 9) and may decide to camp on this cell. In any other case or if other information provided by MIB/SIBs do not satisfy device requirements, the modem can look for other cells in reach, rank them, and provide a list of suitable options to the user device for consideration.

Mentioned second parameter RSRQ is defined by the LTE specification as the ratio of the carrier power to the interference power: essentially this is a signal-to-noise ratio measured using a standard signal. A connection with a high RSRQ should be good, even if the RSRP is low: the modem is able to extract the information in the

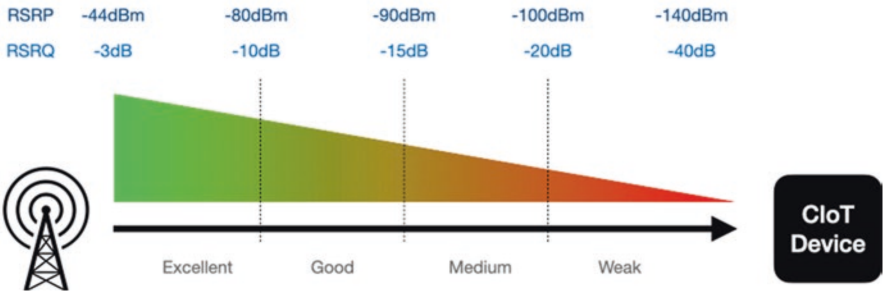


Fig. 9 RSRP and RSRQ—received reference signal power and quality

weak signal because of minimal noise. With RSRP and RSRQ values for all the nearby towers, the modem ranks usable cells for automatic or manual selection.

For IoT application developers using an off-the-shelf modem (see section "NB-IoT Cellular Network Modules" of chapter "Ingredients for NB-IoT Design Concepts") the good news is that they do not have to determine RSRP and RSRQ themselves. Instead, values are continuously being delivered by the modem, they can be read back by the application via proprietary AT commands (e.g., Quectel BG96 is using **AT+QCSQ**). In fact, different modem manufacturers might measure different RSRP and RSRQ values and score them differently, maybe because of their particular product ability how well it can extract signal. Figure 9 provides a rough indication of ranges for RSRP and RSRQ values to be measured at a CIoT device location within cell coverage area:

In manual cell selection mode, the user device host CPU can use these measurement results as an input for independent decision and submit corresponding AT commands in order to tell the modem which cell to camp on.

Radio Resource Control (RRC) and Random Access (RA) Procedures

After installation of an IoT device resp. first power up or relocation from an uncovered area, and a suitable cell has been identified, the device will have to establish a one-to-one communication channel with the eNB base station. Same applies when a user device wakes from power save mode (PSM). For this purpose, it performs a Random Access (RA) procedure to attach to the base station. This establishes a Radio Resource Control (RRC) connection to the base station, allocates resources and schedules for uplink data transmission.

For the radio interface two states are available: **RRC_Idle** and **RRC_Connected** (see Fig. 13). RRC_Connected state is required to receive or transmit data from or to the network. By nature, device power consumption in RRC_Connected state will be significantly higher than in RRC_Idle state, therefore it makes sense to keep

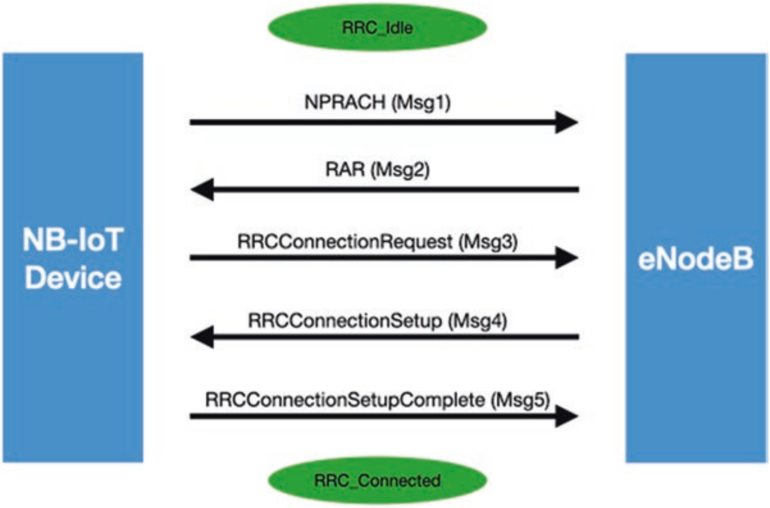


Fig. 10 RRC connection process (simplified)

RRC_Connected periods as short as possible. Once RRC_Connected state is established, only the eNB base station can release this connection and return device to RRC_Idle state. The user device cannot drop the RRC connection other than turning off the radio via AT command interface using the **AT+CFUN=0** command (see also section "Modem Interface"). The base station (eNodeB) has an "inactivity" timer for each module, and if no messages have been exchanged the base station it will release the connection and return RRC status to RRC_Idle. This inactivity timeout usually is configured for expiration after a couple of seconds (e.g., 10 s).

Similar to LTE user devices, NB-IoT devices have to request data transmission service from selected eNodeB base station using a random access (RA) procedure, which is a 5-message handshake (Fig. 10). First, an anonymous signal sequence, called preamble, is sent by the requesting device. The receiving base station then confirms detection of preamble in a message called RA reply (RAR). After RAR reception the device will transmit its unique connection request in Message 3. If Message 3 has been correctly received, base station then replies with Message 4, allowing the respective device to conduct data transmission and to apply RRC_Connected state.

For initial access to an LTE network, dedicated time-frequency resources called Physical Random Access CHannel (PRACH) are allocated periodically by eNB base station. **NPRACH** for NB-IoT is a slightly modified PRACH version in order to address NB-IoT specific characteristics. In fact, NB-IoT networks have to offer an efficient NPRACH procedure to support challenging NB-IoT objectives to manage

- A massive number of ten-thousands of devices covered by a single cell and to
- Access to devices located in areas with weak coverage.

This also means that NPRACH might react to actual service demand and/or to consider cell-specific characteristics of covered area (e.g., increased path loss due to buildings, trees, radio noise), and user devices will have to follow instructions provided by the cell they want to attach to.

The NRACH procedure always starts with a so-called preamble which is indicating an allocation request of a user device to connect. Mentioned SIB2 is periodically broadcasted by the cell and provides NPRACH configuration information which are common to all user devices. Relevant parameters are obtained by the requesting device and used to create an unpredictable NPRACH preamble. Preamble format, max. number of attempts, number of repetitions, etc. of NPRACH are set by requesting device according parameters provided by periodically broadcasted SIB2. SIB2 contains to dedicated random access information blocks called **RACH-ConfigCommon-NB** and **NPRACH-ConfigSIB-NB. RACH-ConfigCommon-NB** is used to specify the generic random access parameters, whereas **NPRACH-ConfigSIB-NB** is used to specify NPRACH configuration for the anchor and non-anchor carriers, for example:

For preparation to start random access procedure (incl. further attempts and repetitions), the device should have a full set of RACH parameters available (see [10], Sect. 5 "Random Access Procedure").

The requesting device needs to determine an appropriate NPRACH configuration according to its coverage class estimation. In general, each network cell specifies its own thresholds for coverage enhancement (CE) level assignment. As mentioned before (see section "Network Selection"), the user device can determine its CE level via measurement of RSRP signal strength. If supported, the cell can specify up to three RSRP thresholds (see **rsrp-ThresholdsPrachInfoList** in Fig. 11) corresponding to four CE levels (i.e., CE0 < Threshold 1 < CE1 < Threshold 2 < CE2 < Threshold 3 < CE3) to be selected by the device according to its coverage class. SIB2 will contain three NPRACH configurations for four CE levels. If the network does not specify any RSRP thresholds, only one single NPRACH configuration is supported, so all devices will have to use it regardless of their actual path loss to the serving base station.

The base station allocates distinct NPRACH for each CE level which occur periodically. To serve up to four NPRACH for the four CE levels, base station divides the 180 kHz bandwidth into 48 subcarriers, each with a subcarrier spacing of 3.75 kHz. The basic unit of the subcarrier allocation for an NPRACH of one CE level is 12 subcarriers. Therefore, an NPRACH for a CE level can have 12, 24, 36, or 48 subcarriers [9]. The requesting device starts a random access procedure by transmitting a preamble in an NPRACH at its CE level which has been determined before. The NB-IoT sequence to be used for the preamble is the same for all devices. Each preamble consists of four symbol groups transmitted without gaps.

Frequency hopping is applied to these symbol groups, i.e., they are each transmitted on different subcarriers (out of allocated group of 12 subcarriers). Among the 12 possibilities, the user device selects the subcarrier for transmission of the first symbol group, if actual RA procedure has been initiated by the device itself. Then, subcarrier selection for next symbol groups is pseudo-random based on an algorithm which has been designed in a way that applied hopping schemes will never overlap and maximize availability of congestion-free preambles (Fig. 12).

Parameter Name	Value options	Short Description	3GPP Reference
NPRACH-ConfigSIB-NB			
nprach-PreambleFormat	fmt0, fmt1, etc.	TDD preamble format	TS 36.211 [14], clause 10.1.6
nprach-Periodicity	40, 80, 160, 240, 320, 640, 1280, 2560 ms	NPRACH resource periodicity	TS 36.211 [14], clause 10.1.6
numRepetitionsPerPreambleAttempt	1, 2, 4, 8, 16, 32, 64, 128	Number of NPRACH repetitions per attempt for each NPRACH resource	TS 36.211 [14], clause 10.1.6
nprach-SubcarrierOffset	0, 12, 24, 36, 2, 18, 34 In number of subcarriers, i.e. offset from subcarrier 0	frequency location of the NPRACH resource.	TS 36.211 [14], clause 10.1.6
nprach-NumSubcarriers	12, 24, 36, 48	Number of sub-carriers allocated to a NPRACH resource	TS 36.211 [14], clause 10.1.6
nprach-NumCBRA-StartSubcarriers	8, 10, 11, 12, 20, 22, 23, 24, 32, 34, 35, 36, 40, 44, 48	Number of start subcarriers from which a UE can randomly select a start subcarrier.	TS 36.321 [15]
nprach-StartTime	8, 16, 32, 64, 128, 256, 512, 1024 milliseconds	Start time of the NPRACH resource in one period	TS 36.211 [14], clause 10.1.6
maxNumPreambleAttemptCE	3, 4, 5, 6, 7, 8, 10	Maximum number of preamble transmission attempts per NPRACH resource	TS 36.321 [15]
npdcch-NumRepetitions-RA	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048	Maximum number of repetitions for NPDCCH common search space (CSS) for RAR, Msg3 retransmission and Msg4	TS 36.321 [15], clause 16.6.
npdcch-StartSF-CSS-RA	1.5, 2, 4, 8, 16, 32, 48, 64	Starting subframe configuration for NPDCCH common search space (CSS), including RAR, Msg3 retransmission, and Msg4	TS 36.321 [15], clause 16.6.
npdcch-Offset-RA	zero, oneEighth, oneFourth, threeEighth	Fractional period offset of starting subframe for NPDCCH common search space	TS 36.321 [15]
rsrp-ThresholdsPrachInfoList	one or two threshold values in dBm	CE level criterion for user device to select a NPRACH resource	TS 36.321 [15]
RACH-ConfigCommon-NB			
powerRampingStep **powerRampingStepCE**	e.g. 2 dB	Power ramping step	TS 36.213 [20] and TS 36.321[15]
preambleInitialReceivedTargetPower	e.g. -104 dBm	PRACH power that eNB expects to receive	TS 36.321 [15]

Fig. 11 NPRACH parameters (selection)

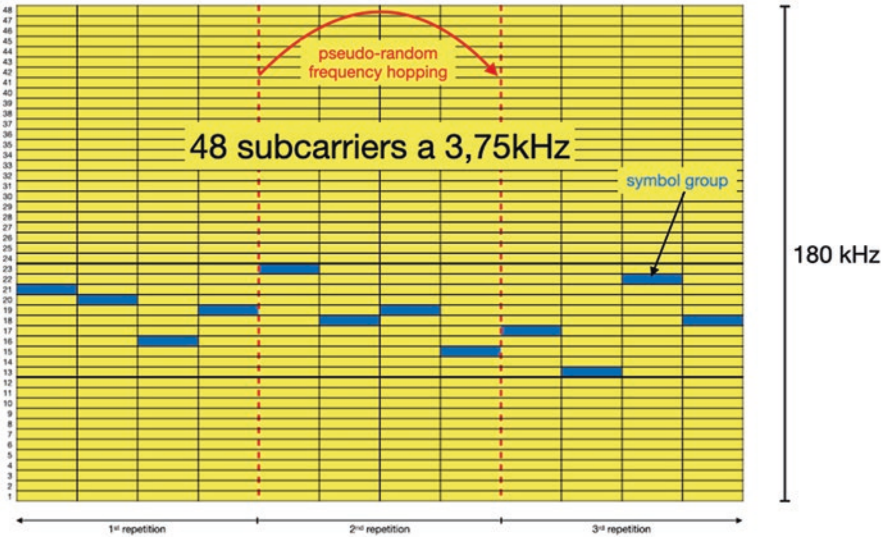


Fig. 12 Preamble sequence (example)

Each symbol group contains one cyclic prefix CP plus five identical symbols, CP length depends one of two available RACH formats determined by SIB2 before. For a transmission attempt, the same preamble is repeatedly transmitted within the same NPRACH (see parameter **nprach-Periodicity** in Fig. 11). The first repetition is transmitted at a subcarrier the user device has chosen from the list of subcarriers allocated to the CE level. For each cell a suitable number of repetitions (up to 128) for different CE level is pre-configured by SIB2 in order to ensure proper reception (see parameter **numRepetitionsPerPreambleAttempt** in Fig. 11). Consequently, NPRACH of the higher CE level has longer duration than that of the lower one. Each time the preamble transmission counter is reaching this parameter, applied CE level is increased. Thus, after the configured number of reattempts in the initially selected CE level, reattempts are performed using NPRACH for a higher CE level—with a higher number of repetitions. If the base station detects an NPRACH preamble, it returns a random access response (RAR), also known as Message 2 (see Fig. 10).

The RAR contains timing and uplink resource allocation as an input to the requesting device for preparation of its RRCConnectionRequest (Message 3). In Message 3 the device will include its identity and indicate its bandwidth demand as well as its capability to support for multi-tone traffic and multi-carrier support. In addition, the device will report its power headroom resp. applied power for preamble transmission P_{NPRACH} vs. maximum transmit power P_{MAX}.

Collision occurs if two or more devices transmitting at the same NPRACH choose the same initial subcarrier. In case of collision, the requesting device has to back off, select a new initial subcarrier and retransmit the preamble in next available NPRACH. This process repeats until the maximum number of attempts in this CE level is reached. The failed device can restart the whole RA procedure in the next higher CE level. The device will declare a RA failure if the RA procedure fails at the highest CE level.

Probability of preamble collisions will increase in congested NB-IoT environments where lots of devices are requesting service. But independent from cell capacity, the individual location of the requesting NB-IoT device resp. its distance to the serving cell is a key aspect for good network coverage and a successful RA process. By definition, the maximum transmit power P_{MAX} of an NB-IoT device is 23 dBm. Other parameters required for calculation of required transmit power P_{NPRACH} are as follows:

1. eNodeB base station receiver sensitivity resp. expected signal strength specified by SIB2 parameter **preambleInitialReceivedTargetPower** (see Fig. 11).
2. Coupling loss of data transmission between NB-IoT device and eNodeB (path loss) with
   (a) SIB2 parameter **referenceSignalPower** (see Fig. 11) plus
   (b) Strength of RSRP reference signal at device location

Example

Let us assume, for example, these input values for P_{NPRACH} calculation:

- **preambleInitialReceivedTargetPower**: −104 dBm
- **referenceSignalPower** = 18 dBm
- RSRP measured by user device: −88 dBm

Calculation:
P_{NPRACH} = **preambleInitialReceivedTargetPower** + Path Loss
$= -104 + 106$
$= 2$ dBm

In this case, applied P_{MAX} of 23 dBm provides sufficient headroom to ensure a seamless random access procedure at lowest coverage enhancement level CE0.

Finally, in Message 4 the network resolves any contention resulting from multiple devices transmitting the same initial RA preamble and provides a connection setup. After reception of Message 4 the device will transit from RRC_Idle state to RRC_Connected mode. The first message of the device in RRC_Connected state is Message 5 RRCConnectionSetupComplete (see Fig. 10).

Later, when a device has returned to the RRC_IDLE state, it may either use again the random access procedure if it has new IoT data to send, or waits until it gets paged. Another option for the device is to store a copy of relevant SIB data and use it after cell reselection or return from out of coverage.

Power Saving Methods

Enabling a significant reduction of overall device power consumption was a major NB-IoT objective. Goal was to enable battery-powered devices with a lifetime of more than 10 years. In fact, besides support for a large number of connected devices (network capacity) and wide/deep network coverage, NB-IoT provides low power modes and power saving mechanisms are key selling points for IoT applications using simple zero-touch and maintenance-free devices a long product lifetime which is not limited to a short battery lifetime.

Of course, promised 10 years battery lifetime will not work for all kind of CIoT devices. Instead, focus is on typical IoT applications which do not require permanent and immediate device availability, e.g., a fluid level meter which is supposed to transmit consumption data only once per day. This kind of IoT device will transmit small amount of data periodically, and afterwards it will stay on hold or idle for the rest of the day. Refer to section "NB-IoT Use Cases" of chapter "IoT Target Applications" for further target applications.

3GPP standardization work has been driven by these IoT-typical characteristics to support implementation an efficient power management strategy for NB-IoT devices. For this purpose, PSM and eDRX power saving modes and other mechanisms are offered to minimize active radio time between the IoT device and the network.

Power Saving Mode (PSM)

PSM is a power saving feature designed for NB-IoT devices which allow to reduce battery power to a minimum. In fact, PSM will switch off the device radio and most part of the network interface which reduces power consumption of this circuit to a few microamps.

But, as usual when leaving RRC_Connected, also during PSM period the IoT device still remains registered in the network, meaning that the device closes the AS connection yet keeps the AS/NAS status, i.e., higher layer connection configuration. Consequently, when the device wants to leave PSM it will just have to resume the previous connection without having to re-attach or re-establish the PDN connections. This avoids extra power consumption due to additional communication with the base station for the higher layer connection establishment procedure.

However, designers will have to keep in mind that—as long as in PSM mode—the device will be unreachable because radio is off, no paging notifications will be received. There are two ways how to wake up the device from PSM state:

1. Event originated from IoT device (e.g., from an integrated sensor) or
2. Expiration of network timer T3412 (TAU or Extended Timer)

T3412 is configured in a way that the user device will wake up periodically to perform a Tracking Area Update (TAU). TAU is a standard LTE procedure used by the device to notify periodically its availability to the network. TAU timer T3412 is similar to LTE but can be programmed with a longer expiration time leading to extremely infrequent periodic TAU events of up to 413 days (!). Expiration of T3412 will end RRC_Idle state and initiate a new or resumed connection.

Besides TAU, in RRC_Connected state all data transmissions between user device and eNB base station will be performed. During this period, the user device

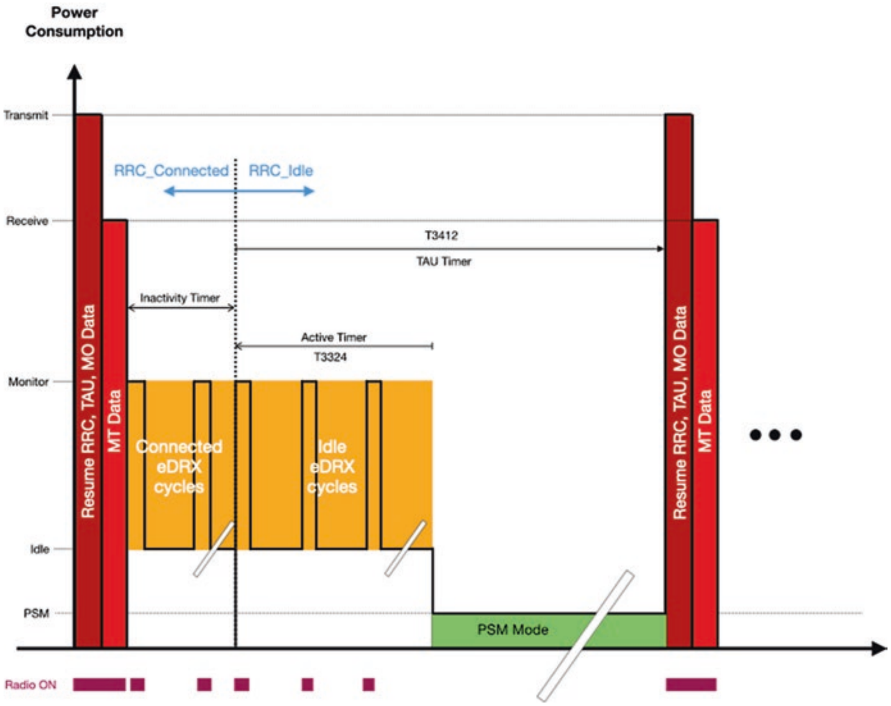


Fig. 13 Low-power modes and timing

listens to paging occasions from the network and will receive queued downlink data (mobile terminated—MT data), if any. And it will perform pending uplink transmissions (mobile originated—MO data), if any. Figure 13 is outlining different operational scenarios of an NB-IoT device and corresponding power consumption levels. Note that power consumption levels correspond to radio activity for different modem activities: transmit data, receive data, listening to paging occasions. For low-power NB-IoT device design it will be crucial to keep these active periods as short as possible, see extra section “Low Power Device Design” of chapter “Designing an NB-IoT Device.”

Extended Discontinuous Reception (eDRX)

Standard LTE power saving feature **DRX** (DRX = discontinuous reception) is designed for efficient support of downlink communication. It can be executed when device is in RRC_Idle mode. In RRC_Idle state, the IoT device cannot transmit data or request resources from the network, but the so-called NPDCCH channel is tracked to determine if there is downlink data pending. During this paging period, energy saving is achieved due to the fact that only some of the subframes are monitored, i.e., the IoT device alternates between active listening of a paging occasion

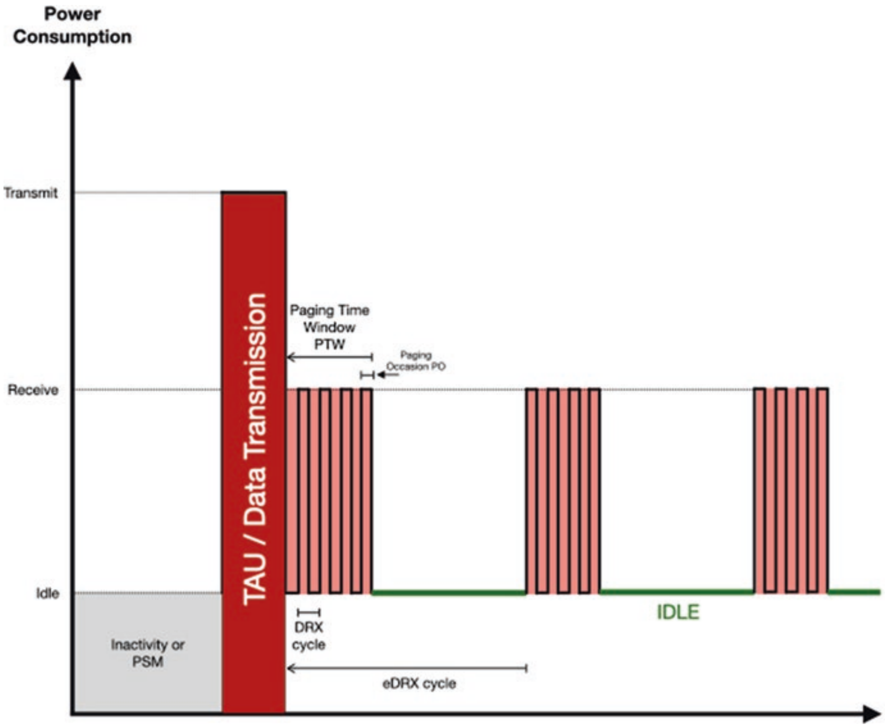


Fig. 14 eDRX timing details

(PO) and sleep for the following paging occasions. This “discontinuous reception” will result in extra latency, but DRX cycle length can be configured by the user in a way that it will not have any negative impact on the IoT application. In RRC_Idle state, DRX cycles of 128, 256, 512, and 1024 radio-frames are supported, i.e., from 1.28 to 10.24 s with a radio-frame of 10 ms each.

The **Extended DRX (eDRX)** feature goes one step further and extends duration of DRX cycles to allow a device to stay in low-power mode after paging window has closed (see Fig. 14), i.e., the IoT device will not listen to control channel for paging messages resp. download traffic until the configured eDRX cycle has ended. If supported by the network, length of this eDRX period is determined by timer T3324 which is programmable by multiples of a hyper-frame (1024 radio-frames = 10.24 s) which results in a maximum duration of ca. 3 h. From IoT device perspective, eDRX is initiated using a dedicated command via AT command interface of the network module (see section “Modem Interface”). Attached network must be able to handle temporary device unavailability and cache network requests until next paging period in RRC_Connected state occurs. This means that requested timing parameters are subject to negotiation, i.e., not confirmed by network yet. If accepted, network will return configured PTW and eDRX cycle values to the requesting device. Based on these agreed timings, a latency-tolerant IoT application can check for pending commands, for example, every hour and apply a corresponding long eDRX cycle afterwards in order to decrease average power consumption to a few milliamps during this period.

Involved timers can be used to adjust timing and schedules of eDRX and PSW cycles according to IoT application requirements, e.g., in terms of availability, latency. This is a summary of relevant conditions and parameters:

Inactivity Timer	Expiration of this timer causes user device change state from RRC_Connected to RRC_Idle. This timer is controlled by the eNodeB and **not** configurable by user device.
TAU Timer (T3412)	This network timer determines after which time a periodic TAU procedure will be performed. Period is user device configurable.
Active Timer (T3324)	This timer is started by the user device when it moves from RRC_Connected to RRC_Idle state. During this period the device runs eDRX cycles, i.e., is reachable for downlink communication. As soon as T3324 expires, the device will enter PSM mode. T3324 is configurable by user device.
DRX cycle	Duration is a multiple of paging occasion (PO) cycles of 1028 ms. User device listens during PO, then sleeps until next PO. This timer is **not** configurable by user device.
Paging Time window (PTW)	This period is representing a paging event, it is composed of several DRX cycles. DRX cycles fit in this PTW window, length is configurable by user device, i.e., number of DRX cycles is also under device control.
PSM mode	PSM is the ultimate device low-power mode, radio is switched off and device is not reachable during PSM period. It is configurable by the user device via T3324.

Based on these configurable parameters, NB-IoT offers are offering a versatile toolbox for application developers which allows adjustment of applied schedules and timings according to specific needs. In particular, use of implemented power saving mechanisms for a long battery life can be balanced with other use case requirements, e.g., maximum allowed latency.

Release Assistance Indication (RAI)

Release assistance indication (RAI) allows the NB-IoT device to indicate to the eNB during RRC_Connected mode that it has no more UL data and that it does not anticipate receiving further DL data. In this case, the device would like the network to release the device to RRC_Idle mode and quickly to save radio power, release assistance indication (RAI) is introduced in Release 13 for Non-Access Stratum (NAS) and in Release 14 for Access Stratum (AS). When AS RAI is configured, UE may trigger a buffer status report (BSR) with size of zero byte as a request to eNB for an **early transition from RRC_Connected to RRC_Idle state**. Without RAI, by default the device would have to wait for the eNB to release the connection via explicit signaling because the eNB is not fully aware of the UE data buffer or the expected DL traffic. Thus, it sends a connection release message that forces the UE to enter idle state after the RRC inactivity timer expires.

RRC Suspend/Resume

This method for signaling reductions allows an NB-IoT device to resume a connection which was previously suspended. As a prerequisite it requires an initial RRC connection establishment (e.g., a periodic TAU) that configures the radio bearers and the Access Stratum (AS) security context incl. security configuration for data protection (e.g., encryptions keys) for an NB-IoT device in the network. Then, the NB-IoT device can enable the RRC connection to be suspended and resumed whenever needed, if the device supports this feature and has been configured accordingly.

When the NB-IoT device transits to RRC Idle state, the RRC Resume procedure will store the connection context and assign a resume ID. This ID is signaled by the RRCConnectionRelease message from the network. Later, when there is new traffic, the device can submit RRCConnectionResumeRequest message containing the associated resume ID to be used by the eNB to access the stored context. On top of resuming a prior connection, the RRC Resume procedure also allows to transmit a small amount of uplink data in Message 5 RRCConnectionSetupComplete (see Fig. 10). For this purpose, payload data is multiplexed with RRC packet data in Message 5 [12].

Preserving the device context instead of releasing it each time it returns to RRC_Idle state means for the device to **skip AS security setup and RRC reconfiguration** for each data transfer, saving a considerable signaling overhead.

Early Data Transmission (EDT)

EDT was introduced in 3GPP Rel 15 and advances benefits of RRC resume procedure which is available also for Rel 13 and Rel 14 NB-IoT devices. EDT allows a device to exchange network data early during random access (RA) procedure in Message 3 resp. Message 4. If EDT is being used during RA procedure, the device can complete its **transmission of user data in RRC_Idle mode** and does not have to transit to RRC_Connected state at all. However, EDT is limited to small data payloads. Uplink EDT supports transport blocks sizes (TBS) of 328 up to 1000 bits (Fig. 15).

Wake-Up Signal (WUS)

In RRC_Idle mode during eDRX cycles the NB-IoT will have to check periodically for paging messages from the network. For this purpose, the device has to activate its baseband receiver and monitor downlink channels NPDCCH (control) and NPDSCH (payload). Unfortunately, at most possible paging occasions (PO) there will be no messages addressed the device, e.g., at nighttime or because unplanned download traffic does not apply to target IoT use case at all. This means that applied radio power for monitoring NPDCCH channel for has been wasted in these cases and reduce battery lifetime without need. In order to increase paging efficiency, a

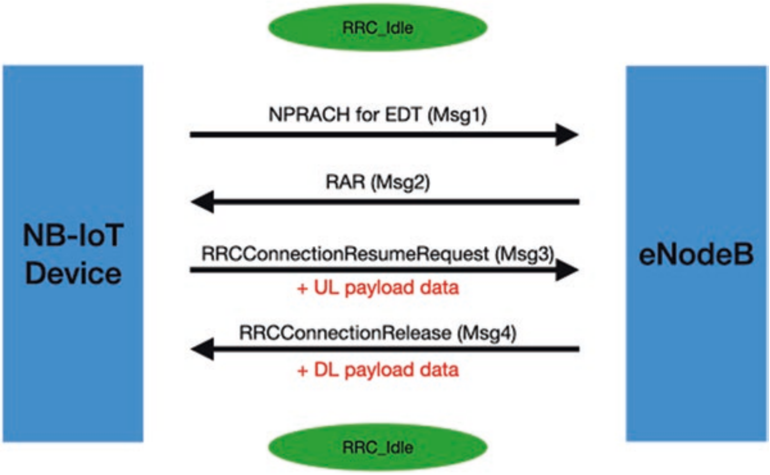


Fig. 15 Early data transmission (EDT)

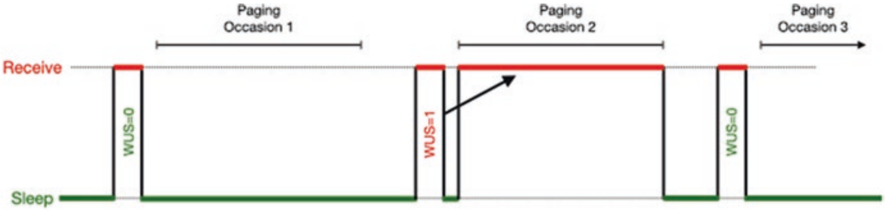


Fig. 16 WUS feature

wake-up signal (WUS resp. NWUS for NB-IoT) was introduced in 3GPP Rel 15. The idea is to **indicate relevance of the following paging occasion** to the NB-IoT device by a single-bit WUS information. WUS takes less time to transmit and is much easier to detect compared to required demodulation and decoding efforts for the full paging event. As a consequence, the NB-IoT device will monitor the NPDCCH channel only in case an WUS signal has been detected before. If no WUS was not set, the device can skip the next PO and stay in power save mode for more time.

Figure 16 illustrates how WUS signal works. Prior to each paging occasion, WUS informs the NB-IoT device the associated paging occasion will indicate pending download traffic for the device or not. If not, i.e., in periods 1 and 3 with WUS = 0, the device will return to sleep immediately. In period 2 with WUS = 1 the device has been alerted and will be ready to monitor and decode NPDCCH channel data accordingly. With the help of WUS, the device can reduce reception periods and save unnecessary power consumption for paging occasions which are not addressed to the IoT device anyway.

WUS is an optional Rel 15 feature requiring support of the device modem and the network as well. On device side, WUS potentially can unfold extra power efficiency if a modem offers a **dedicated low-power WUS detector** to wake up the main baseband receiver in case an WUS was detected. Such a WUS receiver piece of integrated hardware will allow to keep the main receiver switched of during paging as long as the IoT device itself is not addressed [12–14].

Latency

In general, NB-IoT is aiming at latency-tolerant applications and use cases which do not require 24/7 online availability for external parties and/or immediate reaction of the device in case of downlink requests. But nevertheless, one of the NB-IoT objectives was to keep latency for uplink traffic below 10 s. This means that—even under worst case conditions (i.e., MCL = −164 dBm)—the delay to transmit a small data package to the network should not exceed 10 s for a connected NB-IoT device.

During PSM periods, a NB-IoT device is not reachable and first has to wake up before it can transmit any uplink data. This an example how use of power saving features is impacting latency and it shows that both requirements need to be balanced carefully. In addition, NB-IoT devices in poor device coverage conditions assigned with CE1 or CE2 coverage enhancement levels have to repeat message transmissions multiple times, this will reduce data rate and increase latency.

IoT device settings can be controlled with AT commands (see section "AT Command Interface" of chapter "Ingredients for NB-IoT Design Concepts") and some network settings are public (e.g., T3412 TAU Timer), but others are not. For example, an NB-IoT cannot control duration of Inactivity Timer or settings of coverage enhancement thresholds or used TX output power. As a consequence, a network user cannot easily estimate network latency and transmission time. Instead, some simulations and real-world measurements are available to provide more evidence. For example, network infrastructure specialist Ericsson has published some uplink latency numbers [1] for NB-IoT devices at different cell locations at cell border (+0 dB) and beyond (+10 dB and +20 dB):

Coverage (dB)	Sync (ms)	Sys Info MIB/ SIB (ms)	PRACH (ms)	UL transmission (ms)	max. Uplink Latency (s)
+0	340	151	946	167	1604
+10	340	151	1396	1158	3085
+20	520	631	2500	3972	7623

Another paper [15, Table 1] is summarizing best and worst case values for these timing elements. For example, transmission of the PRACH preamble can take 5.6 ms minimum (Format 0, 1 repetition) or 819.2 ms maximum (Format 1, 128 repetitions).

Another report [12, 16] about uplink latency of an 84-byte data package also looks at different NB-IoT operation modes (stand-alone, guard-band, in-band):

Coverage (dB)	Stand-alone (s)	Guard-band (s)	In-band (s)
+0	0.3	0.3	0.3
+10	0.7	0.9	1.1
+20	5.1	8.0	8.1

In this study, transmission time with a stand-alone deployment was shorter compared to in-band or guard-band operation modes because full radio power was allocated to the NB-IoT carrier. More transmission power leads to better coverage and less repetitions. This is speeding up transfer of the final downlink acknowledgement message from the network to the transmitting IoT device. This advantage has minor impact on devices in good coverage conditions because no repeated transmissions are slowing down the overall process. At a coupling loss of less than 144 dB, latency is mostly determined by the time needed to access the network (incl. random access procedure) and to acquire system configuration blocks (MIB, SIB). But at maximum coupling loss 164 dB, latency is dominated by signal repetitions used for coverage extension.

Very obviously, NB-IoT users are facing a significant range of latency effects, mainly caused by assignment of coverage enhancement levels CE0–CE2. For a typical NB-IoT target application (see section "NB-IoT Use Cases" of chapter "IoT Target Applications") it will not matter too much if device-originated IoT data will arrive a few seconds later. But for battery-powered devices, long data transmission periods are critical because they have significant impact on lifetime. For further information about low-power design aspects and how to calculate long-term current consumption see section "Low Power Device Design" chapter "Designing an NB-IoT Device."