

# Adaptive Congestion Control for mMTC in cellular network.

T. Rohan

IIT H (CSE)

3rd May, 2021

# What is mMTC? What is the problem with it?

- mMTC stands for Massive machine type communication.
- It is a type of communication between machines over wired or wireless networks where data generation, information exchange and actuation takes place with minimal or no intervention from humans.
- There is diverse traffic for different commands over various IoT applications, challenging the network efficiency.
- It suffers Congestion and signaling overload due to the high probability of collisions, especially in a dense network scenario.
- Now, we try to propose an adaptive grouping and strategy for the mMTC devices capable of reducing the number of simultaneous transmissions. Hence guarantee the successful transmission of the data in the 5G cellular network.

# mMTC RACH procedure in 5G network

- 5G is the fifth generation of wireless communications technologies supporting cellular data networks.
- The critical enhancements on the cellular network such as eMBB (enhanced Mobile Broadband), mMTC (massive Machine Type Communications), URLLC (Ultra-Reliable Low Latency Communications) characterize 5G.

- ① State of the art of mMTC: The mMTC devices transmit small packet sizes of data following two traffic patterns:
  - ① Periodical : For applications such as Smart grid, periodical traffic pattern is used which is triggered as a Mobile Autonomous Reporting up-link transmission
  - ② Event driven: For applications such as smoke alarm detectors, disaster alert, event driven traffic pattern is used, which is triggered by such emergency events.
- ② RACH Procedure in 5G network : UE stands for User equipment, gNB for gNodeB. (contd. in the next slide)

# RACH Procedure in 5G network

In the current Network scenario, UE performs the random access procedure as the initial contact between UE and the base station.

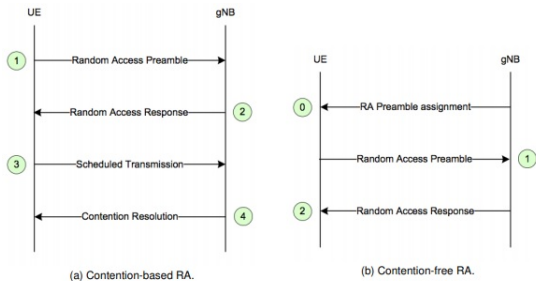


Figure: 5G Random Access Procedure.

- As shown in figure 1, there exists two types of random access procedure: contention-based and contention-free.

# CBRA and CFRA

## CBRA : Contention Based Random access

- Also known as four step RACH Procedure. In this, UE selects a Preamble randomly from a pool of preambles shared with other UE, creating potential risks of selecting the same preamble as another UE and subsequently may experience conflict or contention.
- The gNB uses a contention resolution mechanism to handle this, but, the result is random and not all Random Access succeeds.

## CFRA : Contention Free Random Access.

- Also known as three step RACH procedure. In this, the Preamble is allocated by the gNB and such preambles are known as dedicated random access preamble.
- The dedicated preamble is provided to UE either via RRC signalling or PHY Layer signalling, Thus, ensuring no preamble conflict.

# What is the problem ?

- In the mMTC scenario, to support a tremendous number of devices with zero/very low mobility transmitting occasionally in the network at a low bit rate, high density connection is needed.
- Providing guaranteed network access to these devices presents a challenge for the network operators as the network will suffer for high MAC signaling overhead affecting the mMTC device's energy efficiency characteristic.
- The high number of collisions in the contention-based random access procedure is surely leading to a failure of some mission-critical services.
- Hence the need for an efficient load control mechanism to guarantee successful data transmission.

# Are there any existing approaches?

- To streamline the messaging process.
- Sharing limited resources.
- forming device groups (one of the most effective techniques)
- some physical and medium access techniques

# Our approach

- Our approach uses the application-to device relationship to enhance the grouping mechanism.
- It is done by transforming applications into task executions.
- Our algorithm performs the grouping on the task level. This reduces the signaling overload created as each member of the group.
- In other words, our grouping strategy will reduce the collision rate significantly.



## Problem Formulation.

- ▶ When a set of mMTC devices need to access the network when triggered by some application, they will need to perform the tasks and report to the application server before the expiration time of the application.
  - Any collision or delay during the RACH procedure side will affect the success probability of the data transmission.
- ▶ We aim to group the devices in such a way that all the tasks can be executed within the required time frame, preventing any failure. Hence the need to minimize the total number groups by optimally placing the right device in the group.
- ▶ This formulation is a variation of the 1D bin-packing problem and we use a modified version of the best-fit decreasing (BFD) algorithm to effectively group the MTC devices while respecting latency requirements.

# 1D bin-packing problem

The 1D bin packing problem could be simply put as:

- Given  $n$  items and  $n$  bins, with

$$w_j = \text{weight of the } j\text{th item} \quad (1)$$

$$c = \text{capacity of each bin.} \quad (2)$$

We have to assign each item to one bin so that the total weight of the items in each bin does not exceed  $c$  and the number of bins used is a minimum.

# Best Fit Decreasing Algorithm

- First, we have to sort the items in decreasing order of their weights and assign them indices such that,

$$w_1 \geq w_2 \geq \dots \geq w_n \quad (3)$$

- Now, we assign the current item to the feasible bin (if any) having the smallest residual capacity (breaking ties in favour of the lowest indexed bin). If there exists no such bin which can accommodate a particular item, only then do we introduce a new bin.

# Analogy

- The mMTC devices are analogues to the items in the 1D bin-packing problem.
- The groups which we make are analogues to the bins.
- Latency constraints of the applications represent bin sizes.

# Proposed Grouping Scheme

- ▶ We integrate the scheduling mechanism by sorting in increasing order of the application's time constraint. The grouping process starts by sorting devices in descending order of their weight.
- ▶ For a device, the algorithm evaluates the device-dependent space of each group for all related applications. A device joins a group that has space to accommodate it with minimum remaining space.
- If there is no group with enough space to accommodate the device, a new group is created. It runs the iterations till all devices are grouped.
- ▶ Once grouped, the devices linked to a certain number of tasks in a group are then scheduled in the order of corresponding application's expiration time, so that a task related to the most urgent application can be executed first.
- To prevent any frequent re-grouping, a relaxing factor  $\gamma$  is introduced to control the extension of final bin sizes. The time constraint becomes  $\gamma \times \tau_m$  for further adjustment. The notion of miss rate is then added to the ratio of tasks executed after  $\gamma \times \tau_m$ .

# Terms to know

Symbols	Meaning
$T_i$	Weighting time of $i$ th device
$\tau_m$	Applications' time constraint
$\gamma$	Relaxing Factor
$\gamma \times \tau_m$	Total space of the bin
$\sum_j T_j \times g_{i,j}$	Total space occupied

# Algorithm 1

---

**Algorithm 1** Pseudo-code of the modified BFD Algorithm

---

Sort all devices in decreasing order of their weighting time  $T_n$

$\mathbf{P} = \{\emptyset\}$

**foreach**  $j : j \in \mathbf{N}$  **do**

**foreach**  $m : m \in \mathbf{M}$  **do**

        Find a group  $i \in \mathbf{P}$  that remaining space of group  
         $\tau_m \cdot \gamma - \sum_j T_j \cdot g_{i,j}$  is the minimum

**end**

**if**  $i \neq \{\emptyset\}$  **then**

$j \in i$

**else**

        Create a new group :  $i' \in \mathbf{P} \quad j \in i'$

**end**

**end**

---

Figure: Algorithm 1.

# Performance Analysis

- Let there be 1000 to 5000 smart devices in the simulation setup.
- These devices collect, compute, record, and transfer resource consumption data, on demand, to the company responsible for applications such as monitoring and billing.
- The required number of groups and task missing rates are considered as the major performance indicator.



# Parameter Description and Values

Parameter	Description	Value
$N$	No. of devices	1000 to 5000
$K$	No. of different tasks	10 to 40
	Tasks per application	1 to 5, uniform distr.
	Tasks per device	1, 3, 5, uniform distr.
$M$	No. of applications	20
$t$	Task time	uniform(0.1, 0.2)s
DpG(q)	Device per group with ' $\gamma = q$ '	

# Observations

Devices	DpG(0.3)	Miss	DpG(0.7)	Miss	DpG(1.1)	Miss
1000	7.77	0	17.61	0.28	29.67	5.95
2000	7.14	0	18.35	0.01	27.04	6.00
3000	7.97	0	17.79	0.49	29.22	6.30
4000	7.57	0	18.03	0.54	28.11	5.89
5000	7.30	0	17.40	0.57	28.55	6.66

Table: BFD performance with varying  $\gamma$  in DpG, Tasks = 10

# Observations

- It is observed in the above table that when the packing factor  $\gamma$  increases, the time constraint is relaxed and more devices can be assigned to the groups.
- But, more importantly, our modified BFD algorithm can support massive devices. The gap between the number of devices per group is small regardless of the number of devices (from 1000 devices to 5000).

# Conclusion

► A grouping algorithm has been employed to deal with the congestion problem by finding the best grouping solution to minimize the number of simultaneous transmission in the context of mMTC devices in a 5G network. The obtained results clearly illustrate the grouping performance of our proposed algorithm and its impact on groups by lowering the miss rate probability in each group.

# Original Authors and Details

- Boisguene Rubbens and Chih-Wei Huang  
Department of Communication Engineering, National Central University,  
Taoyuan, Taiwan.
- 2020 IEEE International Conference on Consumer Electronics - Taiwan  
(ICCE-Taiwan)