Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Taleb, Tarik; Bagaa, Miloud; Naas, Si-Ahmed

Scheduled Communications in Next Generation Mobile Networks

Published in:
2018 IEEE Global Communications Conference, GLOBECOM 2018

DOI:
10.1109/GLOCOM.2018.8647344

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Taleb, T., Bagaa, M., & Naas, S.-A. (2018). Scheduled Communications in Next Generation Mobile Networks. In *2018 IEEE Global Communications Conference, GLOBECOM 2018* (IEEE Global Communications Conferences). IEEE. https://doi.org/10.1109/GLOCOM.2018.8647344

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

This is the accepted version of the original article published by IEEE.

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Scheduled Communications in Next Generation Mobile Networks

Tarik Taleb, Miloud Bagaa and Si-Ahmed Naas
Dept. of Communications and Networking, School of Elec Eng, Aalto University, Espoo, Finland
Emails: firstname.lastname@aalto.fi

***Abstract*—$5G$, as the next phase of mobile communications standards, intends to offer connectivity with greater throughput, higher capacity, lower latency and higher mobility range. $5G$ is promised to meet the demands of emerging applications, such as Internet of things (IoT) (e.g., wearables, connected cars, mobile phones, robots, and smart home appliances) to access to the Internet. However, the devices on IoT are expected to grow exponentially in the following years, resulting a dramatic increase in bandwidth requirements. Thus, more efficient management and planning of the network's bandwidth resources are essential in the evolution of mobile systems. In this vein, we propose an Intelligent Scheme (IS) to schedule the communications in the mobile systems for enabling emerging applications, such as connected cars. In our scheme, a core component controller is added to the network, which consists of two parts: the controller database and the controller server. The resource availability status of each cell is recorded in the database. The controller server cooperates with the Intelligent Start Algorithm (ISA) on the end-user controlling the traffic of the whole network. The uploading of time-non-sensitive content is delayed if there are not enough resources at the located cell. We evaluate the performance of this Intelligent Scheme (IS) based on NS-3. The obtained results indicate that our Intelligent Scheme (IS) manages to reduce the packet loss and improve the Quality of Experience (QoE) for users. As most of the added functions can run in the format of software, no dedicated hardware is needed and the overall system cost is expected to be minimal.**

I. INTRODUCTION

In the last several years, cellular network providers have been facing an important challenge consisting in the provisioning of broadband connectivity for connected devices as well as for efficiently serving human-to-human (H2H) communications [1]–[3]. In 2015, there were about 563 million mobile devices which are connected over the Globe. The mobile devices and connections have increased to 7.9 billion in 2015 [4]. According to [5], by 2020 there will be nearly 20.8 billion Internet of Things (IoT) devices connected to the Internet. The bandwidth requirements to accommodate IoT and the connected society have put huge pressure on the existing mobile networks. On the other hand, there is an increasing number of applications in App Stores, such as video streaming applications (e.g., Youtube, Vimeo and iQIYI), voice-over-Internet applications (e.g., Skype and Google Hangouts), social networking services (e.g., Facebook, Twitter, and LinkedIn), map applications (e.g., Google map and Baidu map), and other types of applications (e.g., Paypal and Quora) where the social network platforms are known as one-to-many communication paradigms and the time users spend on these applications is increasing [6]. According to [7], an adult user in the USA spent about 2.8 hours per day on mobile media in 2015. As a result, the mobile data traffic experiences an unprecedented increase. According to [4], the global mobile data traffic was up to 3.7 exabytes per month in 2015, and it will keep upsurging to reach 30.6 exabytes per month in 2020; an eightfold increase over 2015. In order to meet the growing demands and diverse expectations of mobile data users, efficient management and planning of radio spectrum resources are necessary while it is important to start considering customizing mobile telecommunication services and provide a functional differentiation for different user equipments [8]. In this vein, several studies have been carried out, showing the importance of network resource sharing. The Radio Access network sharing is gaining momentum within the different telecommunications standardization bodies. It is supposed to be a crucial part of the next fifth generation (5G) system architecture. Through Network slicing mechanism, a single network infrastructure is shared for multiple network operators while each operator holds its own functionalities and services based on its users' profiles [9]. In [10], the authors provide a detailed survey on the network slicing from end-to-end user perspective while detailing its historical heritage, principal concepts, enabling technologies and solutions.

In LTE, the scheduler determines the resource allocation among users. In Round Robin scheduling strategy, available resource blocks are allocated to the connected users equally. In the Best Channel Quality Indicator (CQI) scheduling method, the E-UTRAN Node B (eNB) assigns resource blocks based on the CQI of channels. However, users having the worse radio link quality (e.g., those located far from the eNB) are unlikely to be scheduled. This algorithm improves the throughput at the expense of poor fairness. In Proportional Fair (PF) scheduling strategy, resource blocks are allocated considering both CQI and fairness. In this paper, we propose an Intelligent Scheme (IS) that allocates resource blocks based on the resource availability of the eNB and the type of the connected user's generated content. We broadly divide user-generated content into two types: time-sensitive content and time non-sensitive content. The time-sensitive content will be assigned resource blocks and will be delivered immediately after their generation. The eNB allocates resources to time-non-sensitive content only if spare resources are available in the located cell. The uploading of time-non-sensitive content will be delayed if there are not enough resources. Two algorithms are designed to retry to deliver the content: Alpha Retry (AR) algorithm and New Cell Retry (NCR) algorithm.

The remainder of this paper is organized as follows. Section II introduces some related work. Section III presents the Intelligent Scheme (IS) in details. Section IV provides the evaluation of the performance of IS based on NS-3 simulations. Section V concludes the paper and illustrates some future research work.

II. RELATED WORK

To cope with the increase in the traffic of cellular networks, some mobile service providers adopt usage-based pricing mechanism and charge heavy data users with special usage fees [11], [12]. As the result shows, one-third of users decrease their Internet usage to avoid additional fees. This approach has also increased providers' profits. However, this approach lowers the Quality of Experience (QoE) for mobile users as they will spend more money on their Internet usage.

A new cellular network architecture is proposed in [13]. It selectively upgrades the infrastructure and increases the capacity of some areas called Drop Zones. There usually exists a delay between content generation and delivery time. The delay differs from several hours to weeks. For example, in the case of Flickr, 55% of content have more than one day lag between content generation till its upload and 25% of content have more than one week lag [13]. Content is usually directly sent to friends by applications, such as WhatsApp and WeChat. In this approach, users do not upload delay-tolerant content until they pass by a drop zone. A greedy algorithm is used to select the candidate base stations as Drop Zones. As the result shows, this scheme can reduce about 24% of the cost spent on building the infrastructure compared to the network architectures that do not adopt such drop-zone mechanism. However, this method does not take the cell's resource availability status into consideration when postponing the delivery of user-generated content.

In addition, a QoE-driven approach is proposed in [14] to optimize the resource allocation for the real-time delivery of content. This approach takes the popularity of the video content into consideration when assigning the resource for videos. Moreover, in [14], the popularity of a video is assumed to follow a Zipf-Mandelbrot law [15]. Popular videos are allocated more bandwidth while less popular ones are allocated less resources. This scheme improved the average user satisfaction by more than 25% compared to the QoE-driven approach without considering video popularity, and 42% compared to the Round Robin scheduling algorithm. However, this service-centric scheme focuses only on the study of video services, and a vast amount of effort has to be put into analyzing the popularity of other types of mobile services. During the mobility of a user, the user's device (UE: User equipment) measures the Reference Signal Received Power (RSRP), the Reference Signal Received Quality (RSRQ) and the Received Signal Strength Indicator (RSSI) of the neighboring cells to perform cell selection and handover. Some resources are spent on these measurement procedures. By prior prediction of the mobility path and the destination of the UE [16]–[18], the scanning time of RSSI can be reduced. Also, the resources can be reserved prior to the handover of the user, so that the delay in resource allocation can be reduced [19]. In the context of path mobility prediction, a mobility prediction scheme based on Markov modeling is performed in [20] to reduce the handover latency. The position of a user after n times of state transition is formulated as follows:

$$p_n = [p][P_{n-1}]^n \tag{1}$$

where p is the initial distribution, P_{n-1} is the current transition probability matrix, and n is the number of state transitions. However, the work in [20] is based on the assumption that p and P_{n-1} are known. It does not specify how to get the initial distribution matrix and transition probability matrix.

A mobility prediction policy called Destination And Mobility path Prediction (DAMP) for mobile networks is proposed in [21]. The DAMP algorithm consists of two schemes: Destination Prediction Model (DPM) for destination prediction and Path Prediction Model (PPM) for path prediction. DPM is based on a belief function Bel and a probability function P. The probability function is derived from the frequently visited locations in the user behavior database. The PPM aims to predict the subsequent of road segments that a user will pass by, on the way to the destination. PPM, which is a semi-Markov process, is the combination of state transition probabilities, the deviation function $r()$, and the penalty function $p()$. As the simulation results show in [21], the DAMP approach outperforms the pattern classification-based scheme in [22], the historical movement-based scheme in [23] and the Markov renewal processes based-scheme in [24]. The mobility-prediction-aware bandwidth reservation (MPBR) policy is presented in [25]. The MPBR consists of three parts: $i)$ handoff time estimation (HTE) scheme to estimate the handoff time, $ii)$ available bandwidth estimation (ABE) scheme to estimate the available bandwidth, and $iii)$ efficient call admission control (ECaC) scheme to manage the resources allocation. The handoff time estimation (HTE) predicts cells to traverse with an estimated reaching time range based on the user's driving behavior database and stopping times database. The available bandwidth estimation (ABE) estimates the available bandwidth of a cell at a given time based on the call data database. The efficient call admission control (ECaC) makes use of handoff time estimation (HTE), available bandwidth estimation (ABE) and an earlier completed call data (ECCD) database to control the bandwidth allocation. The policy manages to reduce hand-off call dropping rate while maintaining an acceptable call blocking rate. However, it mainly focuses on the phone calls.

In this paper, we propose an Intelligent Scheme to schedule communications in mobile networks. A controller is added to the network architecture to control the upload of data so that the overall mobile network resources are most efficiently used and a maximum number of users are fairly served. According to [13], the users are willing to tolerate some delay between the generation and the upload of time-non-sensitive content. Content is broadly divided into two types: $i)$ "one to multiple" content (generated by services such as Facebook, Instagram, and Flickr) that can be delivered in a postponed manner; and $ii)$ "one to one" content (generated by services such as Messenger, WeChat, and Skype). An eNB allocates resources to delay-tolerant content only if there are available resources in this cell. Instant and time-sensitive content will be assigned resources following the underlying scheduler and without any additional delay.

III. INTELLIGENT SCHEME

In this section, we present the main functionalities of our proposed Intelligent Scheme (IS) approach. Specifically, we present $i)$ the Controller; $ii)$ the Intelligent Start Algorithm implemented at the end-user side, and $iii)$ the retry algorithms including Alpha Retry (AR) and New Cell Retry (NCR).

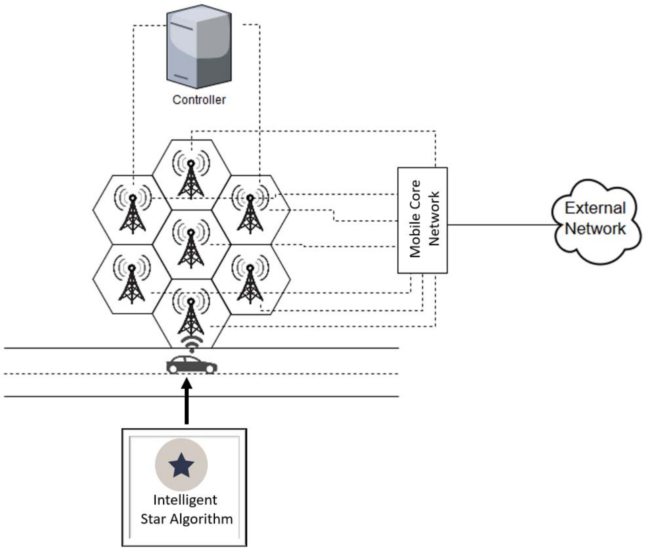

Fig. 1: High level diagram of our proposed architecture for scheduled communications.

A. The controller

In our Intelligent Scheme, a controller is incorporated into the mobile network, as shown in Fig. 1. The controller monitors the status of eNBs and plays a significant role in controlling the uploading pattern of user-generated content. The controller may be directly interfacing to base stations or to the Network Management System (NMS) to collect the statistics required about each base station. Fig. 2 depicts the architecture of the controller. It is composed of two parts, which are the database and the server.

1) The controller database: The controller database records the resource availability status of eNBs. If the resources available at a specific eNB are lower than a predetermined threshold, that eNB is identified as busy. Each eNB is responsible for updating the status records in the controller database. They may connect to and update the database with the following command:

"**Update** cells **set** status = cellStatus **where** id = cellID".

Algorithm 1 Intelligent Start Algorithm.

```
Input:
    remote host,
    $UserList(6-tuple < cur-Cell, servId, servType, rate, destCell, timeGroup >)$
for $node \in UserList$ do
    install ns3::PacketSinkApplication for remote host;
    clientULApps = CreateObject<MyApp>();
    Install clientULApps for User
    std::vector<TimeUnit>::iterator timeItem=(item/>timeGroup).begin()
    if item.$servType$ == 0 then
        while $timeItem! = item- > timeGroup.end()$ do
            Simulator::Schedule(timeItem->startTime,clientULApps)
            timeItem++
        end while
    else
        Simulator::Schedule(timeItem->startTime, intellStart, user, clientULApps, timeItem, (item->timeGroup).size());
    end if
end for
```

2) The controller server: The controller server is responsible for the communication with the end-users. After receiving from a user the ID of the cell the user is attaching to, the controller server checks the resource availability status of the cell from the controller database with the following command: "**Select** status **from** cells **where** id=cellID". The server then replies to the end-user with the cell's current status. The controller cooperates with the Intelligent Start Algorithm on the user side to control the upload data traffic of the network.

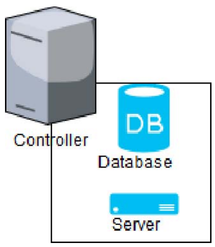

Fig. 2: A simplistic view of the controller's architecture.

B. The intelligent start algorithm

The Intelligent Start Algorithm is implemented at the end-user level. To transmit content, end-terminals first connect to the controller server and consult about the resource availability

Algorithm 2 intellStat function.

```
Input:
      user, app, vectorIterator, vectorSize
 1: Get mob = User Mobility Model
 2: Get Vector <x,y,0> = pos->getPosition()
 3: Get cellID = Current cell of the User
 4: Get status = checkServer(cellID)
 5: if status == 1 then
 6:     start app
 7:     vectorIterator++
 8:     vectorSize–
 9:     if vectorSize > 0 then
10:         Simulator::Schedule(Seconds(nextTime), intellStat, user, clientULApps, vecotrIterator, vectorSize)
11:     end if
12: else
13:     implement Retry Algorithm
14: end if
```

status of the current cell. Algorithm 1 is the pseudocode of the proposed Intelligent Start Algorithm.

Algorithm 1 Line 6 checks the content type. If the type is 0, that means that the content is time-sensitive, the content will be delivered (Line 7 - Line 10) immediately. If the type is 1, which means that there can be a delay between content generation and uploading, then our Intelligent Start Algorithm (Line 12) will be applied.

Function 1: **Simulator::Schedule(timeItem->startTime, intellStart, user, clientULApps, timeItem, (item->timeGroup).size())** schedules function intellStart() to be executed at specific time (timeItem->startTime). user, clientULApps, timeItem anditem-> timeGroup.size() are the parameters passing to itellStart(). Algorithm 2 is the pseudocode for intellStart function.

Function 2:**checkServer(cellId)** (Line 4) opens a TCP socket and connects to the controller server. It then requires the resource availability status of the located cell. If the returned value equals to 1 (Line 6), then the user transmits the content immediately. If it equals to 0, which means that there are not enough resources at the located cell, the immediate uploading of the content will be blocked and Retry Algorithm is implemented (Line 13). Line 10 is for users who use a service for multiple times.

C. Retry algorithms

In this paper, we envision two variants of the retry algorithm. The first is the Alpha Retry algorithm, the second is the New Cell Retry algorithm. In the case of Alpha Retry, the user retries to transmit the content Alpha time later. In the New Cell Retry algorithm, the user retries to transmit the content when it enters into a new cell. Algorithm 3 illustrates the process of the New Cell Retry algorithm.

The position vector *pos* of the user is obtained from its mobility model (Line 1). Line 2 gets the located cell *curCell* from user position. If the user enters into a new cell and this cell has available resources, the content will be transmitted immediately. Otherwise, it executes the New Cell Retry Algorithm periodically (Line 11). Lines 5-9 represent a scheduling process in the case of uploading multiple contents.

Algorithm 3 New Cell Retry algorithm.

```
Input:
      user, app, vectorIterator, vectorSize
 1: Get Vector <x,y> pos = Current position of user
 2: Get curCell = Current located cell
 3: if currCell == previousCell && checkServer(curCell) == 1 then
 4:     start app
 5:     vectorIterator++
 6:     vectorSize–
 7:     if vectorSize > 0 then
 8:         Simulator::Schedule(nextTime, &intellStat, user, clientULApps, vecotrIterator, vectorSize)
 9:     end if
10:
11: else if previousCell != destinationCell then
12:     Simulator::Schedule(Second(1), &newCellRetry, user, clientULApps, vecotrIterator, previousCell, destinationCell vectorSize)
13: end if
```

IV. PERFORMANCE EVALUATION

The performance evaluation is carried out based on computer simulations using NS-3, which is an open source and discrete-event network simulator based on C++ and Python. In this section, Alpha Retry (AR) Algorithm stands for Intelligent Scheme with Alpha Retry, New Cell Retry (NCR) Algorithm stands for Intelligent Scheme with New Cell Retry.

A. Default parameters

Table I presents the default parameters used in our NS-3 simulations. In the simulations, eNB allocates the available resource blocks (RB) among all active flows, and the Resource Block Per Flow (RBPF) is used to identify the resource availability status of a cell. If RBPF is lower than a threshold, the respective eNB is considered as busy. The default threshold is set to 12 (2.4 MHz) while the uplink bandwidth of the eNBs is set to 25 (5 MHz).

One dimension to evaluate the performance of our Intelligent Scheme is the Cumulative Packets Loss (CPL). The NS-3 uplink RLC trace periodically calculates the transmitted packets TP, received packets R_P, transmitted data in bytes $T_{b,t}$, and received data in bytes $R_{b,t}$. The CPL $A_{L,t}$, at time t, is defined as follows:

$$A_{L,t} = T_P - R_P + A_{L,t-1} \tag{2}$$

where $A_{L,t-1}$ represents the cumulative number of lost packets at time (t-1). It is a recursive process. The other dimension to evaluate the performance of the proposed Intelligent

TABLE I: Default parameters of the NS-3 simulations.

UL Bandwidth	25 (5MHz)
LTE user Phy TxPower	10 dB
LTE eNB Phy TxPower	30 dB
Handover Algorithm	ns3::A2A4RsrqHandoverAlgorithm
Scheduler	ns3::RrFfMacScheduler
Point-to-Point Channel	MTU 1500
Point-to-Point Channel Data Rate	1 Gbps
Point-to-Point Channel Delay	10 ms
A Cell Area	300 * 300 m^2
Alpha	2.5 hours
Threshold	12 (2.4 MHz)
Simulation Time	10 hours

Scheme is Loss Ratio (LR). It measures the loss ratio in bytes. The LR at time t ($L_{R,t}$) is defined as follows:

$$L_{R,t} = A_{L,b,t}/A_{T,b,t} \tag{3}$$

where $A_{L,b,t}$ denotes the cumulative data loss in bytes at time t. $A_{T,b,t}$ is the cumulative number of transmitted data in bytes at time t. $A_{L,b,t}$ and $A_{T,b,t}$ are defined as follows:

$$A_{L,b,t} = T_{b,t} - R_{b,t} + A_{L,b,t-1} \tag{4}$$

$$A_{T,b,t} = T_{b,t} + A_{T,b,t-1} \tag{5}$$

where $T_{b,t}$ is the amount of transmitted data in bytes during time (t-1) to the amount of transmitted data in bytes during time t. $R_{b,t}$ denotes the received data in bytes during time (t-1) to the received data in bytes during time t. $A_{T,b,t}$ and $A_{L,b,t}$ are recursive processes.

B. *Results analysis*

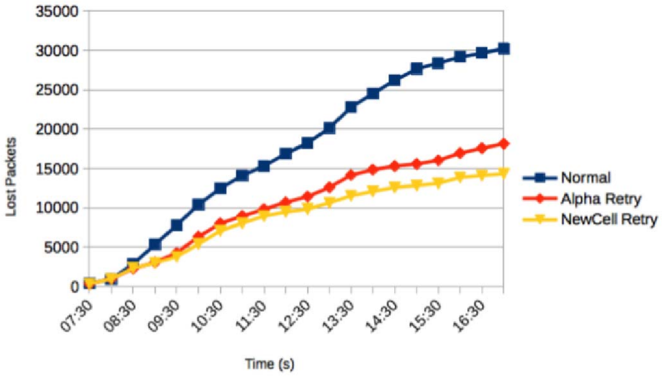

Fig. 3: The CPL of Three Algorithms

We compare the performance of the Intelligent Scheme (in case it implements AR and NCR) against a baseline scheme where no intelligence is used to admit, delay, or block the upload of generated content (Normal algorithm). Figs. 3 and 4 plot the Cumulative Packets Loss (CPL) and Loss Ratio (LR) of the Normal algorithm, AR and NCR algorithms.

As Fig. 3 shows, the number of packet losses experienced under the Normal Algorithm do not differ much from those experienced in case of the Alpha Retry and New Cell Retry algorithms and that is before 8:30 am. For example, at 8:00 am, the CPL of the Normal algorithm is about 942, compared to 958 for both AR and NCR algorithms. This is mainly because, at the beginning of the simulation, the cells are not busy and the content is delivered without delay. However, as time goes by, more and more users connect to the network and the network traffic increases rapidly. The resource availability status of some cells consequently change. In our Intelligent Scheme, the upload of delay-non-sensitive content is blocked in cells that do not have enough resources. As a result, the CPL differences between the Normal Algorithm and the other two Intelligent Start Algorithms increase significantly. At the end of the simulation, the CPL of Normal Algorithm reaches 30241, exceeding that of AR Algorithm and NCR Algorithm by 12088 and 30241, respectively. We can observe from Fig. 3 that the gap between the LR values of AR Algorithm and NCR Algorithm occurs at 9:30 am. The Alpha Retry Algorithm loses more packets than the New Cell Retry Algorithm but the losses are still less than in the case of the Normal Algorithm. As Fig. 4 illustrates, at 7:30 am the LR of Normal Algorithm is higher than the other two algorithms. It can be explained by the fact that, at this time, the $A_{T,b,t}$ of the three algorithms is small. A small difference in $A_{L,b,t}$ may lead to a relatively great difference in $L_{R,t}$. As $A_{T,b,t}$ increases, the LR differences between Normal Algorithm and the other two algorithms decrease. However, after 8:30 am, the differences enlarge. It is due to the fact that the $A_{L,b,t}$ of Normal Algorithm is much higher than the other two algorithms. At 10:00 am, the LR of Normal Algorithm reaches the peak with 29.2%, compared to 24.3% for Alpha Retry and 22.8% for New Cell Retry. As time goes on, the difference stays around 5%. The LR differences of Alpha Retry Algorithm and NCR Algorithm are slight. In general, the LR of the NCR Algorithm is lower than that of the Alpha Retry Algorithm. At 10:00 am, the gap reaches the peak. The LR of NCR is 22.8%, while that of AR is 24.3%. Our scheme provides a smaller CPL and a lower LR. It can be concluded that our Intelligent Scheme (both Alpha Retry Algorithm and New Cell Retry Algorithm) outperforms the Normal Algorithm.

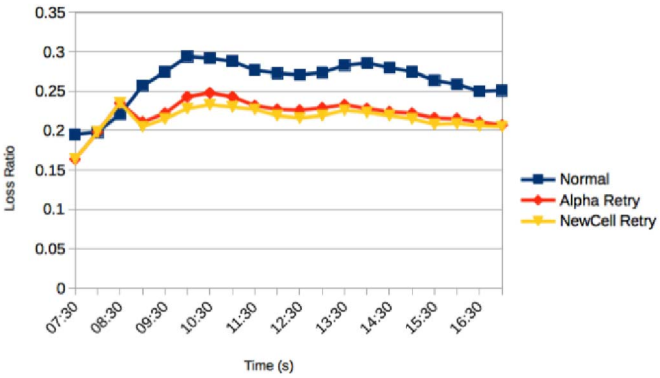

Fig. 4: The LR of Three Algorithms

V. CONCLUSION

In this paper, we have proposed an Intelligent Scheme to schedule the communications in a mobile network. The objective is to relieve the huge pressure on the wireless network when users connect to the network. In order to overcome the resource limitation, our proposed approach effectively manages the allocation of eNB resources and service admission. The added controller implements the service admission. It makes the decision on whether to allow the immediate upload of the time non-sensitive content based on the resource availability status of the cells. Cooperated with the Intelligent Start Algorithm, the controller changes the users uploading data pattern. The RBPF is used to identify the status of the cell. The contents are categorized according to their utilization and characteristics. For the retry of content upload, we have proposed two algorithms Alpha Retry (AR) and New Cell Retry (NCR). We also compared the performance of our scheme with the Normal Algorithm, a policy without any intelligence in service admission. The results show that our proposed scheme (both AR Algorithm and NCR Algorithm) reduce the cumulative packets loss (CPL) significantly while decreasing the loss ratio (LR) of the network. The simulation results demonstrate that this Intelligent Scheme outperforms the Normal Scheme, especially case of a heavy payload network.

As future work, we plan to implement a user dedicated application based on the proposed Intelligent Scheme that controls the upload of delay-tolerant content. Before the uploading of those delay tolerant content, the application checks the available bandwidth from the controller and decides whether to deliver the content or not. Moreover, a sophisticated destination prediction algorithm, a path prediction algorithm based on machine learning, and an available bandwidth estimation algorithm will be integrated into the scheme. Idle cells that a user may pass can be predicted. Therefore, our application can indicate to the users where and when to upload their time non-sensitive content. Whilst the performance evaluation was conducted based on NS3 in this paper, as future research work, we intend using more realistic scenarios mimicking the behavior of mobile users in terms of mobility and mobile service consumption and that is leveraging our Mobile Service Usage Cartography tool [26], developed by the MOSA!C Lab Research Group[1]. We also plan to devise and evaluate different variant algorithms for retrying the upload of time-non-sensitive content.

ACKNOWLEDGMENT

This work was supported in part by the Academy of Finland Project CSN under Grant No. 311654.

REFERENCES

[1] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jntti, and A. Ghosh, "Machine-type communications: current status and future perspectives toward 5g systems," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 10–17, September 2015.

[2] T. Taleb and A. Kunz, "Machine type communications in 3gpp networks: potential, challenges, and solutions," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 178–184, March 2012.

[3] A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, "Cellular-based machine-to-machine: overload control," *IEEE Network*, vol. 26, no. 6, pp. 54–60, November 2012.

[4] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 20152020 white paper," February 2016.

[5] Gartner, "Gartner says 6.4 billion connected things will be in use in 2016, up 30 percent from 2015," November 2015.

[6] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, "Coping with emerging mobile social media applications through dynamic service function chaining," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2859–2871, April 2016.

[7] D. Chaffey, "Mobile marketing statistics compilation," April 2016.

[8] T. Taleb, B. Mada, M. Corici, A. Nakao, and H. Flinck, "Permit: Network slicing for personalized 5g mobile telecommunications," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 88–93, May 2017.

[9] I. Afolabi, M. Bagaa, T. Taleb, and H. Flinck, "End-to-end network slicing enabled through network function virtualization," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, Sept 2017, pp. 30–35.

[10] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing & softwarization: A survey on principles, enabling technologies & solutions," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[11] M. Hamblen, "At&t moves closer to usage-based fees for data," December 2009.

[12] T. Taleb, N. Nasser, and M. P. Anastasopoulos, "An auction-based pareto-optimal strategy for dynamic and fair allotment of resources in wireless mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4587–4597, Nov 2011.

[13] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming user-generated content in mobile networks via drop zones," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 2840–2848.

[14] A. E. Essaili, E. Steinbach, D. Munaretto, S. Thakolsri, and W. Kellerer, "Qoe-driven resource optimization for user generated video content in next generation mobile networks," in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 913–916.

[15] M. E. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary physics*, vol. 46, no. 5, pp. 323–351, 2005.

[16] I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing service replication for mobile delay-sensitive applications in 5g edge network," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[17] I. Farris, T. Taleb, A. Iera, and H. Flinck, "Lightweight service replication for ultra-short latency applications in mobile edge networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[18] I. Farris, T. Taleb, H. Flinck, and A. Iera, "Providing ultra-short latency to user-centric 5g applications at the mobile network edge," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, 2017.

[19] T. Taleb and K. B. Letaief, "A cooperative diversity based handoff management scheme," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1462–1471, April 2010.

[20] S. H. Ariffin, N. Abd, N. E. Ghazali *et al.*, "Mobility prediction via markov model in lte femtocell," *International Journal of Computer Applications*, vol. 65, no. 18, 2013.

[21] A. Nadembega, A. Hafid, and T. Taleb, "A destination and mobility path prediction scheme for mobile networks," *IEEE transactions on vehicular technology*, vol. 64, no. 6, pp. 2577–2590, 2015.

[22] T. Anagnostopoulos, C. Anagnostopoulos, and S. Hadjiefthymiades, "Efficient location prediction in mobile cellular networks," *International Journal of Wireless Information Networks*, vol. 19, no. 2, pp. 97–111, 2012.

[23] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen, "Path prediction and predictive range querying in road network databases," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 19, no. 4, pp. 585–602, 2010.

[24] H. Abu-Ghazaleh and A. S. Alfa, "Application of mobility prediction in wireless networks using markov renewal theory," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 788–802, 2010.

[25] A. Nadembega, A. Hafid, and T. Taleb, "Mobility-prediction-aware bandwidth reservation scheme for mobile networks," *IEEE Trans. Vehicular Technology*, vol. 64, no. 6, pp. 2561–2576, 2015. [Online]. Available: https://doi.org/10.1109/TVT.2014.2345255

[26] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: Vnf placement algorithms for a dynamic amp;amp; realistic edge cloud environment," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

---

[1] MOSA!C Lab Research Group - www.mosaic-lab.org