Resource Allocation for Loss Tolerant Multicast Video Streaming

Sadaf ul Zuhra*, Student Member, IEEE, Prasanna Chaporkar*, Member, IEEE, and Abhay Karandikar*, Member, IEEE

Abstract—The massive explosion of video traffic coupled with the growing number of high-speed mobile connections has transitioned us from the age of downloads to an era of streaming. Video streaming often requires transmitting the same content to a large audience simultaneously. Multicast provides an efficient means of catering to such services. However, in multicast, the base station cannot transmit content at rates higher than that decodable by users with the worst channels. This makes the system performance dependent on the weakest users in the system. In this work, we propose a method to address this problem. Videos can tolerate some packet loss without significant degradation in the quality experienced by users. We leverage this property to propose a loss tolerant video multicasting system that allows for some controlled packet loss while meeting the quality requirements of users. We convert the problem of resource allocation in this system into the problem of stabilizing a virtual queueing system. We propose loss optimal policies for this system that successfully meet the loss requirements of all users. Since the proposed policies are designed primarily for video streaming services, we use traces from actual video streams to generate realistic video traffic patterns in all our simulations.

Index Terms—Multicast, Video streaming, LTE, 5G, MBMS, Resource allocation.

1 INTRODUCTION

Global Mobile Data Traffic (MDT) is expected to grow at a compound annual growth rate of 46 percent from 2017 to 2022, reaching 77.5 ExaBytes per month by 2022 [1]. Video traffic is the largest contributor to this massive amount of mobile data. Videos are projected to account for 79% of the total MDT by 2022 [1]. With the widespread deployment of Fourth Generation (4G) Long Term Evolution (LTE), the number of high-speed mobile connections has seen an enormous increase. This has also contributed to an unparalleled amount of videos being sent over the Internet every day.

This explosion of video traffic has transitioned us from the age of downloads to an age of streaming. This paradigm shift has been primarily driven by the growing popularity of platforms like Netflix, YouTube, Hulu, Amazon Prime Video. Prevalence of such streaming services has led to a fundamental shift in the way users consume online video content. Users increasingly prefer streaming content over cellular networks on the go on mobile devices like smartphones. This often involves online streaming of television (TV) programs, live streaming of major world events, sports matches. All these applications require transmitting the same content to a large audience simultaneously. Using unicast transmissions for such applications consumes a substantial fraction of the limited amount of spectrum available for use by cellular systems. This has created an immediate need for techniques that can better utilize the system bandwidth and accommodate this surge in video traffic within the available spectrum. Multicast transmission is one such technique [2], [3]. Using multicast, several users can receive the same content over shared resources. Multicast services in LTE and Fifth Generation (5G) communications are termed as Multimedia Broadcast Multicast Services (MBMS).

To serve all User Equipments (UEs) in a multicast group, data cannot be transmitted at a rate greater than what can be decoded by the weakest UE in the group. This decreases the system throughput and impacts the performance of other services in the cell. Moreover, UEs with good channel states are constantly forced to settle for lower rates despite their high Channel Quality Indicator (CQI) values, leading to user dissatisfaction. These problems can be overcome by exploiting the loss tolerant nature of video streams. Videos can tolerate packet losses as high as 40% [4]. For an H.264/AVC encoded video, decoders like FFmpeg and JM can conceal as much as 39% packet loss with no deterioration in the quality of video observed by the user [4]. This can be leveraged to build video specific resource allocation policies which can significantly reduce the bandwidth consumption of video streams. Loss tolerant nature of video streams has not been exploited for performance improvement of multicast in the existing literature.

In this work, we propose a loss tolerant model for video streaming that allows for some controlled packet losses. In this model, UEs have a certain tolerance for loss which is leveraged by the system to selectively drop some packets. We design efficient resource allocation algorithms for such a system and evaluate their performance through extensive simulations. Allowing for some controlled losses allows the flexibility of not having to serve all UEs in each sub-frame. The transmission rates in some sub-frames can, therefore, be higher than what can be decoded by the weakest UEs. This results in higher throughput and better user satisfaction. In the next section, we discuss some relevant literature.

1.1 Related Literature

In the existing literature, no resource allocation policy takes advantage of this unique loss tolerant nature of video streams for optimizing resource allocation for multicast services. However, various forms of source coding have
been developed that make video streams more resilient to losses [5], [6], [7]. In [6], [7], the authors design variations of Distributed Source Coding (DSC) frames for reducing error propagation, facilitating view switching and minimizing the effect of packet loss in Interactive Multiview Video Streaming (IMVS). IMVS enables users to switch between different views of a video stream. The authors design Drift Elimination DSC (DE-DSC) frames that halt error propagation within the video stream due to the dependence of frames on their predecessors.

The problem of grouping and resource allocation for multicast streaming has been studied by us in [2] and [3]. The objective of resource allocation in [2], [3] is to satisfy all the multicast UEs while minimizing the number of Physical Resource Blocks (PRBs) used in doing so. In [8], the authors propose a Frequency Domain Packet Scheduler (FDPS) for MBMS that maximizes the minimum rate achievable by UEs in a PRB. It uses a conservative approach in that it only minimizes the damage caused by the worst PRB assignment. In [9], the authors propose a fair and optimal resource allocation for MBMS. It is assumed that the video content is simultaneously available through unicast and MBMS and the primary problem seeks to jointly optimize over the grouping of UEs and allocation of resources to unicast and MBMS. The resource allocation scheme proposed in the paper allocates resources to groups proportional to the number of UEs in the group.

Resource allocation for MBMS Operation On-Demand has been studied in [10]. The authors consider Quality of Experience (QoE) metrics such as user engagement instead of Quality of Service (QoS) metrics like throughput and loss as the utility functions to be maximized by the resource allocation schemes. All the video streams are assumed to be encoded using Scalable Video Coding (SVC). In [11], the authors have used convex optimization to obtain an optimal solution for multicasting Dynamic Adaptive Streaming over HTTP (DASH) [12] and SVC streaming content over LTE. The problem optimizes the Modulation and Coding Scheme and the Forward Error Correction code rates used while allocating resources.

In [13], the authors use a pricing based scheme for allocating resources to multicast groups streaming SVC video content. Users are divided into three multicast groups based on the price they pay. UEs that pay the most receive the maximum number of enhancement layers. In [14], the authors investigate the use of Random Network Linear Coding (RNLC) for improving the performance of multicast services. They use two different forms of RNLC for multicasting H.264/SVC videos in a generic cellular system. The authors in [15] deal with optimizing the delivery of network coded SVC content using MBMS. They make use of Unequal Error Protection (UEP) for ensuring reliability of the multi-layer video transmission. They propose a UEP Resource Allocation Model (UEP-RAM) that provides a much better coverage than conventional multi-rate transmission [16].

In [17], the authors propose a scheduling scheme for MBMS broadcast that is focused on reducing the average latency of packets in the system. The proposed scheme starts transmission in unicast mode and gradually moves to broadcast mode as the number of UEs increases. In [18], the authors deal with efficient broadcasting in LTE using MBMS. The proposed broadcasting mechanism has been given the name of Broadcast over LTE (BoLTE). Their resource allocation algorithm uses a water filling form of proportional fair scheduling [19], [20]. The authors have evaluated the performance of BoLTE using a WiMAX testbed.

The existing literature on multicast does not take advantage of the loss tolerant nature of video streams. In this paper, we consider the use of MBMS primarily for video streaming applications. Since video streams can handle losses as high as 40% without significantly altering the quality perceived by the user [4], we exploit this property to design efficient resource allocation policies for MBMS. A similar approach has also been used in [21] to design a regular service guarantee algorithm for a wireless network with a number of links, only one of which can transmit in a time slot. In our system, the degree of packet loss that can be tolerated by a user depends upon factors like the video being streamed and the channel quality experienced by the user. Allowing for some losses can help in better resource utilization and in controlling congestion in the system during peak traffic hours. It also reduces the dependence of a multicast group on the UE with the worst channel quality as the resource allocation policy is no longer constrained to serve every UE in every sub-frame.

In most of the existing multicast literature, the rate achievable by a group is assumed to be the same in all PRBs. This assumption significantly simplifies the resource allocation problem. Without the channel variability over PRBs, all PRBs are equivalent for a group and the problem reduces to determining the number of PRBs to be allocated to a multicast group. We, however, take into account the fact that, due to fast fading, the channel state and hence the CQI for a group or user may vary over different PRBs in a single sub-frame as well. Therefore, in addition to determining the number of PRBs to be allocated to a group, the identity of the PRBs to be allocated must also be specified.

1.2 Contributions

The main contributions of this paper are summarized below:

- We propose a loss tolerant MBMS system for multicast video streaming that exploits the loss tolerant nature of videos to improve the performance of multicast streaming. The proposed system allows for some controlled packet losses while satisfying the quality requirements of users.
- We convert the problem of resource allocation in loss tolerant MBMS systems to the problem of stabilizing a fictitious virtual queueing system. We prove that stabilizing the token queues in the constructed queueing system is equivalent to satisfying the loss requirements of the users.
- We propose two loss optimal policies for resource allocation in loss tolerant MBMS systems, namely, Loss Optimal Resource Allocation (LORA) and priority LORA (p-LORA). These policies do not require any statistical information of the channel states of users. Channel states can vary arbitrarily and can also be correlated across users. The proposed policies are optimal in the sense that they can satisfy the loss requirements of all UEs whenever any other policy, including offline policies with complete information of channel states of users, can do so.

p-LORA provides an improvement over LORA in terms of the burstiness of the losses encountered by the users. It
ensures that no user is starved for long periods at a stretch, thus ensuring a better quality of experience. We also propose an algorithm for efficient polynomial time implementation of these policies.

- We conduct extensive simulations to evaluate the performance of the proposed policies. Since these policies are designed for video streaming, we use traces from actual videos [22], [23] to simulate realistic video traffic patterns.

The rest of this paper is organized as follows. We discuss the system model the problem formulation in Sections 2 and 3 respectively. The construction of the virtual queuing system and related results are presented in Section 4. In Section 5, we present the proposed resource allocation algorithms and their polynomial time implementation. The simulation results are given in Section 6 and we conclude in Section 7. In the interest of preserving the flow of the paper, proofs of all the theorems and lemmas are given separately in Section 9.

2 SYSTEM MODEL

We model an MBMS system consisting of an LTE cell in which L different MBMS videos are being streamed. Our system consists of an LTE cell with L different MBMS services. There are M UEs in the cell that can subscribe to any of these services. Let [n] = {1, ..., n} and let |A| denote the cardinality of a set A. Thus, [M] and [L] denote the set of UEs and the set of multicast groups, respectively. UEs subscribed to the ith video stream form multicast group Gi and we use i(k) to denote the index of the group to which UE k belongs. The number of UEs in Gi is denoted by Ki. Each MBMS group is allocated one PRB in each sub-frame. The ith MBMS service requires data to be transmitted to its subscribers at rate Rk. Whenever a PRB is allocated to multicast group Gi, data is transmitted in that PRB at the corresponding rate Rk. For every MBMS stream, a data packet arrives at the beginning of each sub-frame and is transmitted in the same sub-frame.

A resource allocation policy Γ decides which PRB will be allocated to which group in each sub-frame. We define an allocation vector BΓ[t] for policy Γ in sub-frame t. BΓ[t] is a vector of length L which specifies which PRB, if any, has been assigned to each group. Note that Γ is completely defined by the value of BΓ[t] in each sub-frame t. We use BΓ[t] to denote the ith entry of vector BΓ[t]. If Gi is not scheduled for reception in sub-frame t, then BΓ[t] = 0, otherwise BΓ[t] takes the value of the PRB number allocated to Gi.

We allow the channel states of UEs to vary across time and frequency. As a result, channel experienced by a UE varies from one sub-frame to another and also across PRBs within a sub-frame. There is a certain maximum rate rkj[t] that UE k can successfully receive in PRB j of sub-frame t [24]. This rate is a function the CQI experienced by the UE in that sub-frame. As a result, a UE may not receive the MBMS content successfully even after a PRB has been assigned to its multicast group. When a UE successfully receives data in a sub-frame, we say that the UE has been served in that sub-frame. Note that a UE being scheduled and being served is not the same. We distinguish between these two terms below:

- We say that a UE has been scheduled in a sub-frame if a PRB is allocated to its group in that sub-frame. For instance, UE k ∈ Gi is said to have been scheduled in sub-frame t under policy Γ if BΓ[t] = j ≠ 0.
- We say that a UE has been served in a sub-frame if it has been scheduled in that sub-frame and is able to successfully decode the received content. For instance, UE k ∈ Gi, is said to have been served in sub-frame t under Γ if BΓ[t] = j ≠ 0 and Rk ≤ rkj[t].

For each MBMS video stream, a packet arrives at the beginning of each sub-frame. Therefore, if a UE is not served in a sub-frame, it experiences a packet loss. We denote the loss encountered by UE k under policy Γ in sub-frame t by ℓk[t]. For k ∈ Gi and BΓ[t] = j ≠ 0, we have:

\[ \ell_k[t] = \begin{cases} 0, & \text{if } R_{i(k)} \leq r_{kj}[t], \\ 1, & \text{otherwise.} \end{cases} \]  

For BΓ[(k)] [t] = 0, the UE is not scheduled for reception and so, ℓk[t] = 1.

Since all UEs in the system are streaming video content, they can tolerate some packet loss without significant degradation in the quality of video seen by them. This loss tolerance may differ from UE to UE depending upon the channel conditions experienced and the video resolution chosen by them. A higher resolution would typically mean a lower loss tolerance and vice versa. We use ℓk to denote the fractional loss that can be tolerated by UE k. ℓ̃ = [ℓ1, ..., ℓM] denotes the loss tolerance vector for the system.

A compressed video stream is made up of Group of Pictures (GoP). A GoP comprises a series of Intra-coded (I), Predicted (P) and Bidirectional predicted (B) frames. I frames are self-contained and do not require other frames to be decoded. P frames are dependent on their preceding I frames for being correctly decoded, and B frames are dependent on both preceding and following I and/or P frames for being correctly decoded. Due to this, it is difficult to estimate the impact of loss of I and P frames on the video quality [22]. Therefore, we assume that all I and P frames of the videos are transmitted without any loss and we use loss tolerant transmission only for sending the B frames. For sending I and P frames, the eNB can use lossless allocation policies [3] to allocate sufficient resources so that these frames are transmitted without any loss. In the next section, we formally define the resource allocation problem for this loss tolerant MBMS system.

3 PROBLEM DEFINITION

We begin by stating some important definitions that will be used in formulating the problem.

Definition 1. Feasible resource allocation: Resource allocation in a sub-frame is said to be feasible if it assigns at most one PRB to each multicast group such that no two groups are assigned the same PRB. In other words, a feasible resource allocation in sub-frame t corresponds to an allocation vector B[t] such that no two non-zero elements of the vector are equal i.e., if BΓ[t] = j ≠ 0, then BΓ[t] ≠ BΓ[t] for every i′ ≠ i.

Definition 2. Feasible resource allocation policy: A feasible resource allocation policy Γ is a policy that chooses a feasible allocation vector in each sub-frame.
A resource allocation policy can make use of the knowledge of current channel states of UEs, the allocation information of the previous sub-frames, the loss tolerance of UEs and the losses encountered by UEs in the past to make allocation decisions in a sub-frame. It could even be an offline policy that could make allocation decisions in advance with the knowledge of the channel conditions of all sub-frames including the ones in the future.

**Definition 3.** Average packet loss: We denote the average packet loss encountered by a UE \( k \) under resource allocation \( \Gamma \) by \( \bar{\ell}_k^T \). It is the total packet loss per unit time and can be mathematically expressed as follows:

\[
\bar{\ell}_k^T = \lim_{T \to \infty} \sup_T \frac{1}{T} \sum_{t=1}^{T} \ell_k^T[t].
\]

**Definition 4.** Feasible region of a policy: The feasible region of a resource allocation policy \( \Gamma, \mathcal{L}^\Gamma \), is the set of all loss tolerance vectors, \( \mathbf{\ell} \) that can be satisfied by \( \Gamma \) i.e. \( \mathbf{\ell} > \mathbf{\bar{\ell}} \) with probability (w.p.) 1. Here, \( \mathbf{\bar{\ell}} = [\bar{\ell}_1^T, \ldots, \bar{\ell}_M^T] \).

**Definition 5.** Feasible region of the system: The feasible region of the system is the set of loss vectors \( \mathcal{L} = \bigcup_{\Gamma} \mathcal{L}^\Gamma \) where the union is over all feasible \( \Gamma \).

**Definition 6.** Optimal policy: The optimal resource allocation policy \( \Gamma^\ast \) is a policy whose feasible region is the set of loss vectors \( \mathcal{L}^{\Gamma^\ast} = \bigcup_{\Gamma} \mathcal{L}^\Gamma \).

Our objective here is to determine the optimal resource allocation policy \( \Gamma^\ast \). We design this optimal policy using results from queueing theory. Towards this end, we convert the resource allocation problem in a loss tolerant MBMS system to the problem of stabilizing a queueing system.

### 4 A Virtual Queueing System for Resource Allocation

We convert the problem of optimal resource allocation in a loss tolerant MBMS system to the problem of stabilizing a virtual queueing system. Towards this end, we first discuss the construction of the virtual queueing system.

![Virtual queueing system model](image)

**Fig. 1: Virtual queueing system model**

#### 4.1 Construction

We construct a virtual queueing system that is maintained at the eNodeB as shown in Fig. 1. It consists of fictitious queues corresponding to each user. The length of the queue of a user is an indicator of how much loss a user has encountered. The state of this queueing system is completely described by the lengths of these virtual queues. Arrival and departure processes of these queues is defined below. Note that these queues are virtual and no physical entities arrive or depart from them. Arrivals and departures merely represent an increase or decrease in queue lengths. We use the term ‘token’ to refer to the virtual entities that make up these queues and refer to these queues as ‘token queues’.

The arrival process for the token queue of \( k \)-th UE is denoted by \( \{\lambda_k[t]\}_{t \geq 1} \). \( \lambda_k[t] \) is a binary random variable indicating arrival of a token to \( k \)-th queue in sub-frame \( t \), and has the expected value \( \lambda_k = 1 - \bar{\ell}_k \). This value of the expected arrival rate will mean that stabilizing this virtual queue will ensure that, in the actual system, UE \( k \) is served in more than \( 1 - \bar{\ell}_k \) of the sub-frames. Arrivals across sub-frames are assumed to be independent and identically distributed. Across users, the arrival processes are assumed to be independent. \( \lambda = [\lambda_1, \ldots, \lambda_M] \) denotes the system arrival rate vector.

We define another indicator random variable \( \mu_k[t] \) that indicates whether or not UE \( k \) has been served in sub-frame \( t \) under \( \Gamma \). \( \mu_k^T[t] = 1 \) if and only if (iff) \( k \) is served in sub-frame \( t \). If \( B_{i(k)}[t] = j \), then, \( \mu_k[t] = 1 \) iff \( j \neq 0 \) and \( B_{i(k)} \leq r_{kj}[t] \). Otherwise, \( \mu_k[t] = 0 \). Let \( Q_k^T[t] \) denote the length of queue \( k \) at the beginning of sub-frame \( t \) under \( \Gamma \). Note that

\[
Q_k^T[t + 1] = \max\{Q_k^T[t] + \lambda_k[t] - \mu_k[t], 0\}.
\]

Now, the stability region of the queueing system thus constructed can be defined as follows:

**Definition 7.** Stability region of the queueing system: The queueing system is said to be stable if the expected queue lengths stay finite for every queue i.e. \( \sup_t \mathbb{E}[Q_k[t]] < \infty \) for every \( k \). A resource allocation policy that stabilizes the system is called a stable resource allocation policy. The stability region of a resource allocation policy \( \Gamma \) is the set of arrival rate vectors for which the system is stable under \( \Gamma \). The stability region of the queueing system is the union of the stability regions of all feasible \( \Gamma \)’s. We denote it as \( S \).

**Definition 8.** Throughput optimality: A resource allocation policy \( \Gamma \) is said to be throughput optimal if \( \Gamma \) can stabilize the queueing system if some policy can do so. This means that if the queueing system is at all stabilizable, \( \Gamma \) will stabilize it.

Note that eNodeB knows the loss requirements of UEs as well as their channel states, so, it has all the information needed to maintain the virtual queueing system. In the next section, we examine the stability region of the constructed queueing system and relate it to the feasible region of the optimal resource allocation policy.

### 4.2 Feasible Region of the Optimal Resource Allocation Policy and Stability Region of the Queueing System

In this section, we prove that stabilizing the constructed virtual queueing system is equivalent to meeting the loss requirements of the multicast UEs. This will establish the equivalence of the stability region of the constructed queueing system and the feasible region of the optimal resource allocation policy. We begin by defining a few terms.

Define a set \( \mathcal{B} = \{B_1, \ldots, B_{|\mathcal{B}|}\} \) containing all possible PRB allocation vectors to \( L \) groups. The cardinality of this set \( |\mathcal{B}| = \binom{L}{N} \times L! \). In LTE, channel states are quantified
in terms of CQI values. According to Third Generation Partnership Project (3GPP) standards [24], a total of 15 CQI values are defined in LTE. Since the number of CQI values is finite, the possible channel states of UEs can take finitely many values. We define a set $C$ that contains all possible channel state combinations of all UEs in the system. For an LTE system, $C$ will therefore be a set of $15^M$ CQI vectors, each of size $M$. Let $g$ be the probability distribution over $C$. That is, the channel state of the system in a sub-frame $t$, $C(t) = C$ w.p. $g(C)$. We denote by $\mu_{B,C}$, the vector of service rates of UEs corresponding to allocation $B_i$ in CQI state $C \in C$. Note that $\mu_{B,C}$’s are binary vectors of size $M$. We use $\mu = \{\mu_{B,C}\}_{B \in B}$ to denote the set of possible service rate vectors in channel state $C$. For a given $C \in C$, define a distribution $w_C = \{w_{B,C}\}$ over the set of $\mu_{B,C}$’s where $w_{B,C}$ denotes the probability of choosing allocation $B_i$ in channel state $C \in C$. Using these definitions, we now define the following Linear Program (LP):

$$LP(\delta) : \sum_{C \in \mathcal{C}} \sum_{B \in \mathcal{B}} g(C)w_{B,C}\mu_{B,C} = \lambda + \delta,$$

$$w_{B,C} \geq 0 \ \forall \ B_i \in \mathcal{B}, \ C \in \mathcal{C},$$

$$\sum_{B \in \mathcal{B}} w_{B,C} = 1, \ \forall \ C \in \mathcal{C},$$

where $\delta$ is a non-negative real number. Note that $\{w_C\}_{C \in \mathcal{C}}$ are the variables in this LP. Denote by $\Lambda(\delta)$ the set of arrival rate vectors $\lambda$ such that the feasible region of $LP(\delta)$ is non-empty. Define two sets, $\Lambda^o = \bigcup_{\delta > 0} \Lambda(\delta)$ and $\overline{\Lambda} = \bigcup_{\delta \geq 0} \Lambda(\delta)$. In the next result, we establish the relation between $\Lambda^o$, $\overline{\Lambda}$ and stability region of the queueing system $\mathcal{S}$. This result is essential for relating the feasible region of the optimal resource allocation policy to the stability region of the queueing system.

**Theorem 1.** $\Lambda^o \subseteq \mathcal{S} \subseteq \overline{\Lambda}$.

**Proof.** The detailed proof is given in Section 9.1.

From here onwards, we consider $\Lambda^o$ to be the stability region of the queueing system. We now state and prove the following important theorem that relates the feasible region of the optimal resource allocation policy to the stability region of the queueing system.

**Theorem 2.** The loss requirement of a UE is met iff its token queue in the queueing system is stable. Therefore, the feasible region of the optimal allocation policy $\Gamma^e$, $\mathcal{L}^{\Gamma^e}$ is equivalent to the stability region of the queueing system, $\mathcal{S}$, i.e. $\hat{\ell} \in \mathcal{L}^{\Gamma^e}$ iff $(1 - \hat{\ell}) \in \mathcal{S}$. Here, $\hat{1}$ is a vector of ones of same size as $\hat{\ell}$.

**Proof.** The detailed proof is given in Section 9.2.

We have now established that the stability region of the virtual queueing system is same as the feasible region of the optimal resource allocation policy $\Gamma^e$. Henceforth, we do not explicitly consider meeting the loss requirements of UEs. Instead, we focus our attention on stabilizing the token queues corresponding to each UE knowing that, stabilizing the token queues of UEs will ensure that their respective loss requirements are met. In the next section, we propose policies for resource allocation in loss tolerant MBMS systems.

## 5 Resource Allocation Algorithms for Loss Tolerant Multicast Streaming

In the loss tolerant MBMS system under consideration, each UE has a certain loss threshold that needs to be met. UEs are satisfied as long as losses encountered by them are kept within their acceptable thresholds. In this section, we propose loss optimal resource allocation policies that can meet the loss requirements of all UEs in the system.

### 5.1 Loss Optimal Resource Allocation (LORA)

LORA makes scheduling decisions in a sub-frame $t$ based on the token queue lengths of users $Q_k[t]$’s. For ease of notation, we use $\Gamma_0$ to denote LORA\(^1\). In each sub-frame $t$, $\Gamma_0$ chooses service vector $\mu^{\Gamma_0}[t]$ according to the following optimization problem:

$$\mu^{\Gamma_0}[t] = \arg\max_{\mu \in \mathcal{S}} \sum_{k=1}^{M} Q_k[t] \mu_k^{\Gamma_0}[t],$$

where $\mu_k^{\Gamma_0}[t]$ is the service rate of UE $k$ in sub-frame $t$ under $\Gamma_0$. $\Gamma_0$ maximizes the sum of the queue lengths of the UEs served in sub-frame $t$. We have already established in Section 4 that stabilizing the token queues ensures that the loss requirements of UEs are met. Therefore, to prove that $\Gamma_0$ can successfully meet the loss requirements of multiscat UEs, it is sufficient to show that $\Gamma_0$ stabilizes the virtual queueing system. We prove this in the following result.

**Theorem 3.** For any stabilizable arrival rate vector $\lambda$, $\Gamma_0$ stabilizes the queueing system.

This theorem implies that as long as the system is stabilizable, i.e., there exists some policy $\Gamma$ that can stabilize the queueing system, so can $\Gamma_0$. Note that $\Gamma$ is not restricted to using the same information that is available to $\Gamma_0$. $\Gamma$ could be using information of the past and future allocations and channel conditions to make allocation decisions. We claim that, despite this, $\Gamma_0$ will successfully stabilize the system using only the knowledge of the current state of the queueing system to make the scheduling decisions.

**Proof.** The detailed proof is given in Section 9.3.

We now have a loss optimal policy that meets the loss requirements of users by making allocation decisions based on UE token queue lengths. However, in addition to the amount of packet loss in a video stream, we would also like to control the pattern in which these losses occur. Even if a user has a high tolerance for loss, we would like to avoid large number of consecutive packet losses. Starving users for a large number of consecutive sub-frames may lead to user dissatisfaction and result in users leaving the multicast session. Therefore, a loss tolerant resource allocation policy should also restrict the amount of consecutive packet losses encountered by a UE in addition to the long term average packet loss. We propose such a policy in the next section. This policy ensures that users do not remain unserved for long periods at a stretch which leads to better loss performance, reduced burstiness of packet losses, and improved user satisfaction.

\(^1\)The names of policies (e.g., LORA) and their symbols (e.g., $\Gamma_0$) are used interchangeably throughout this paper.
5.2 Priority LORA (p-LORA)

p-LORA also makes scheduling decisions in sub-frame $t$ based on the queue lengths $Q_k[t]$'s in that sub-frame. However, in p-LORA, we use an additional priority vector to increase the probability of serving a previously unselected queue. We use $\Gamma_P$ to denote p-LORA. In every sub-frame $t$, $\Gamma_P$ chooses service vector $\mu^{\Gamma_P}[t]$ according to the following optimization problem:

$$\mu^{\Gamma_P}[t] = \arg \max_{\mu^{\Gamma_P}[t] \in \mathcal{M}} \sum_{k=1}^{M} (Q_k[t] + (c_k[t] + 1) \times s) \mu_k^{\Gamma_P}[t],$$

where $\mu_k^{\Gamma_P}[t]$ is the service rate of UE $k$ in sub-frame $t$ under $\Gamma_P$, $c_k[t]$ is the priority weight ascribed to the token queue of UE $k$ and $s$ is a positive constant. $c_k[t]$ is defined as follows:

$$c_k[t] = \begin{cases} 0, & \text{if } \mu_k[t-1] = 1, \\ \min(c_k[t-1] + 1, \kappa), & \text{otherwise}. \end{cases}$$

$\kappa$ is the maximum value that the priority weights can take. Also, $c_k[0] = 0, \forall k$. We use $c[t] = [c_1[t], \ldots, c_M[t]]$ to denote the vector of priority weights of all the queues in sub-frame $t$. Increasing $c_k[t]$ increases the contribution of UE $k$ in (3) which increases its likelihood of being served under $\Gamma_P$.

When using policy $\Gamma_P$ for resource allocation, the state of the queueing system can be completely defined by the queue lengths and the value of the priority counter of each queue. We denote the state in sub-frame $t$ under policy $\Gamma_P$ by $Q^{\Gamma_P}[t] = [Q_1^{\Gamma_P}[t], \ldots, Q_M^{\Gamma_P}[t], c[t]]$. Since scheduling decisions under $\Gamma_P$ in a sub-frame are based only on the state of the system in that sub-frame, the evolution of states of the system form a Discrete Time Markov Chain (DTMC). In the next result we prove that this DTMC is countable, irreducible and aperiodic.

**Lemma 1.** The DTMC formed by the evolution of the states under $\Gamma_P$, $Q^{\Gamma_P}[t] = [Q_1^{\Gamma_P}[t], \ldots, Q_M^{\Gamma_P}[t], c[t]]$ is countable, irreducible and aperiodic.

**Proof.** The detailed proof is given in Section 9.4. \qed

We now prove that $\Gamma_P$ is throughput optimal, i.e., $\Gamma_P$ will stabilize the queueing system if any other policy can do so.

**Theorem 4.** For any stabilizable arrival rate vector $\lambda$, $\Gamma_P$ stabilizes the queueing system.

**Proof.** The detailed proof is given in Section 9.5. \qed

In the next section, we present a generalization of the Exponential (Queue length) rule (EXP-Q) which was proposed in [25]. The EXP-Q rule is a well known throughput optimal policy for scheduling multiple flows over a time varying wireless channel that minimizes the maximum delay encountered in the system [26]. We use it as a benchmark for performance evaluation of the proposed policies. The rule, however, considers that there is a single channel that can be used by one flow at a time. Therefore, we propose a generalization of EXP-Q for use with multicast transmission and multiple channels.

5.3 Generalized Exponential (Queue length) rule ($\Gamma_E$)

EXP-Q rule is a throughput optimal policy [25] that schedules a single queue $k$ in a time slot $t$ such that:

$$k \in \arg \max_k \gamma_k \mu_k[t] \exp \left( \frac{a_k Q_k[t]}{\beta + [Q_k[t]]^{\eta}} \right),$$

where $\mu_k[t]$ is the rate of service of queue $k$ in sub-frame $t$, $a_k$, $\gamma_k$ and $\eta$ are constants and $Q[t] = (1/N) \sum_k a_k Q_k[t]$. The EXP-Q rule is designed for use in a system where a single time varying channel is shared by multiple flows. We generalize the EXP-Q rule to include multicast transmission and multiple channels (in the form of PRBs) available for scheduling multiple multicast and unicast flows. In the existing form, the EXP-Q rule cannot be used for resource allocation in such a system. Therefore, we modify it as follows.

We use $\Gamma_E$ to denote the modified EXP-Q rule. Since we have multiple channels available and multiple groups can be scheduled for service in a sub-frame, the policy has to determine an allocation vector $B^{\Gamma_E}[t]$ instead of choosing a single entity to be scheduled in a sub-frame. As defined in Section 3, $B^{\Gamma_E}[t]$ is a vector that specifies which PRB is allocated to which multicast group. We define $\Gamma_E$ as the policy that chooses service vector $\mu^{\Gamma_E}[t]$ according to the following optimization problem:

$$\mu^{\Gamma_E}[t] = \arg \max_{\mu^{\Gamma_E}[t] \in \mathcal{M}} \sum_{k=1}^{M} \gamma_k \mu_k^{\Gamma_E}[t] \exp \left( \frac{a_k Q_k[t]}{\beta + [Q_k[t]]^{\eta}} \right),$$

where $\mu_k^{\Gamma_E}[t]$ is the service rate of UE $k$ in sub-frame $t$ under $\Gamma_E$, $\Gamma_P$ chooses the service vector that maximizes (5). It then determines the allocation vector $B^{\Gamma_E}[t]$ corresponding to the service vector $\mu^{\Gamma_E}[t]$. $\Gamma_E$ can also be used for joint allocation of resources to unicast and multicast transmissions.

5.4 Computational Complexity

All the resource allocation policies discussed in this section have a brute force computational complexity of $O(MN^2)$. This makes them unsuitable for use in practical systems unless we can find efficient means of implementing them. It turns out that these policies can be implemented in polynomial time using a Maximum Weight Bipartite Matching (MWBM) [27]. We discuss the details of this implementation in the next section.

5.5 Polynomial Time Implementation

We make use of MWBM for an efficient polynomial time implementation of the resource allocation policies proposed in this section. The MWBM brings down the computational complexity of their implementation to $O(NL^2)$. The policies can thus be implemented in polynomial time. We begin with the construction of the underlying bipartite graph which is the same for all the policies except for the edge weights which are different for each policy. We discuss the implementation for $\Gamma_0$ in detail. The procedure and proof involved can be directly used for $\Gamma_P$ and $\Gamma_E$ as well with modified edge weights. The modifications involved will be specified at the end of this section.

Construct a bipartite graph $G = (U, V, E)$ where vertex set $U$ is the set of $L$ multicast groups and vertex set $V$ is the
set of $N$ PRBs. We define the service rate of a UE $k \in G_i$ in PRB $j$ in sub-frame $t$ as follows:

$$
\nu_k^j[t] = \begin{cases} 
0, & \text{if } R_i > r_k^j[t] \\
1, & \text{otherwise.} 
\end{cases}
$$

The weight of an edge connecting $i \in U$ to $j \in V$ is $w_i^j[t] = \sum_{k \in G_i} Q_k[t] \nu_k^j[t]$. The resulting bipartite graph is illustrated in Fig. 2. A MWBM of $G$ that matches each vertex in $U$ to a unique vertex from $V$ results in an allocation equivalent to $\Gamma_0$. We prove this in the following result.

**Fig. 2:** Bipartite graph for implementation

**Lemma 2.** Maximum weight bipartite matching for graph $G$ results in resource allocation according to policy $\Gamma_0$.

**Proof.** The detailed proof is given in Section 9.6.

The same MWBM can be used for implementing $\Gamma_p$ and $\Gamma_E$ by changing the edge weights. For $\Gamma_p$ we will have:

$$
w_i^j[t] = \sum_{k \in G_i} (Q_k[t] + (c_k[t] + 1) \times s) \nu_k^j[t]. 
$$

(6)

For $\Gamma_E$ the edge weights will change to:

$$
w_i^j[t] = \sum_{k \in G_i} \gamma_k \nu_k^j[t] \exp \left( \frac{a_k Q_k[t]}{\beta + (Q_k[t])^\eta} \right).
$$

(7)

The same proof as in Lemma 2 follows to show that the MWBM for graph $G$ with the edge weights defined in (6) and (7) results in the implementation of policies $\Gamma_p$ and $\Gamma_E$ respectively. In the next section, we present the results of the simulations performed for evaluating the performance of the proposed resource allocation schemes.

### 6 Simulations

We study the performance of the proposed allocation algorithms in an LTE MBMS system. We simulate an LTE cell with UEs distributed uniformly at random through the cell. There are $L$ MBMS video streams available in the cell and each UE is subscribed to one of these streams. UEs subscribed to the same MBMS service form a multicast group and receive the relevant content on common PRBs. We use the MATLAB [29] based LTE simulator designed in [30] for these simulations. To create LTE specific physical layer conditions, we create channels using the models recommended by 3GPP [28]. SNR to CQI and CQI to rate mappings have been done according to 3GPP specifications [28]. Other relevant simulation parameters are given in Table 1.

For each stream, a packet arrives at the beginning of a sub-frame and is transmitted in the same sub-frame at the required rate. Each UE can tolerate some amount of packet loss. The loss tolerable by a UE depends on the quality of video required by it and the channel conditions it's experiencing. We observe the packet loss encountered by UEs under the proposed policies and compare their performance with that of the modified EXP-Q rule.

Since the proposed policies are meant for use with video streaming services, we use traces from actual videos to generate realistic video traffic patterns in these simulations. These video traces have been obtained from the video trace library of Arizona State University (http://trace.eas.asu.edu/) [22], [23]. The videos used are Silence of the Lambs, Star Wars IV, Tokyo Olympics, NBC News and Sony Demo. The videos are all H.264/AVC encoded with a GoP size of 16 with 15 B frames in each group.

Since I and P frames are needed to decode other frames in a GoP, we ensure that all I and P frames are transmitted without any loss. We use the proposed lossy allocation policies only for sending the B frames. This is a recommended practice in network simulations with video traces [22] since it's difficult to estimate the impact of loss of I and P frames on video quality [22]. For sending I and P frames, we allocate sufficient resources to the streams and transmit at the rate corresponding to the weakest UE to ensure that those frames are successfully received by all UEs.

The plots in Fig. 3 compare the losses encountered by UEs to their loss tolerances. For these plots, we run the simulations for the entire duration of all 5 videos and then average the results. Fig. 3a and 3b show this comparison for LORA and p-LORA respectively. Both these policies succeed in meeting the loss requirements of all the UEs. The virtual queueing system is, thus, stable under both the proposed loss optimal policies. Fig. 3c plots the losses encountered under the modified EXP-Q rule. We observe that several UEs experience losses significantly greater than their tolerable limits and the queueing system is rendered unstable.

**Fig. 4a** compares the plots of the average losses encountered by UEs under the three schemes. For this, losses encountered per second have been exponentially averaged for each UE. Every point in the plot is then obtained by averaging over all the UEs. We observe that the EXP-Q rule results in a better loss performance than LORA. Even though EXP-Q gives us a better average loss performance, as we have observed in Fig. 3, it fails to meet the loss requirements of several UEs. On the other hand, despite a

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>System bandwidth</td>
<td>20 MHz</td>
</tr>
<tr>
<td>eNB cell radius</td>
<td>150 m</td>
</tr>
<tr>
<td>Path loss model</td>
<td>$L = 128.1 + 37.6 \log(10d)$, $d$ in kilometers</td>
</tr>
<tr>
<td>Lognormal shadowing</td>
<td>Log Normal Fading with 10 dB standard deviation</td>
</tr>
<tr>
<td>White noise power density</td>
<td>$-174 \text{dBm/Hz}$</td>
</tr>
<tr>
<td>eNB noise figure</td>
<td>5 dB</td>
</tr>
<tr>
<td>eNB transmit power</td>
<td>46 dBm</td>
</tr>
<tr>
<td>Number of PRBs</td>
<td>100 per sub-frame</td>
</tr>
</tbody>
</table>
greater average system packet loss, LORA meets the loss requirements of all the UEs. p-LORA results in the least average packet loss among the three schemes. Fig. 4b illustrates the PSNR degradation due to packet loss encountered under the three policies. PSNR degradation is calculated as the difference between the PSNR of the transmitted and the received videos. We observe that EXP-Q leads to the most degradation in PSNR. LORA and p-LORA result in a significantly less loss in PSNR of the received video streams.

As discussed in Section 5, in addition to the amount of packet loss, the pattern in which the losses occur also has a major bearing on user experience. While some amount of loss spread more or less uniformly through a session may lead to no degradation in quality, concentrated packet losses can be extremely annoying in a video stream and may even lead to UEs quitting the session. To observe the temporal pattern of losses encountered under the three policies, we plot the percentage packet loss pattern of a UE with high loss tolerance, under heavy traffic conditions. This is plotted as a function of time in Figure 4c. EXP-Q results in the most variable loss pattern. The losses per second see jumps as high as 10% from one second to another. LORA does better than the EXP-Q rule. p-LORA provides the most uniform loss pattern of the three. It, therefore, controls the burstiness of the losses encountered and ensures that no UE is starved for long periods at a stretch.

These simulation results clearly establish the effectiveness of the proposed policies. The use of traces of actual videos further strengthens the case for using loss tolerant allocation policies for streaming video content.

7 Conclusions

Videos streams can tolerate a certain amount of packet loss without significant degradation in the quality perceived by the end users. In this paper, we leverage this property to improve the performance of multicast video streaming in MBMS. We consider an MBMS system where users can tolerate a certain amount of packet loss depending on the type of video they are streaming and their channel quality. We address the problem of resource allocation in such a system. We construct a fictitious virtual queueing system to represent the actual loss tolerant MBMS system. We convert the optimal resource allocation problem for the said system to the problem of stabilizing the constructed virtual queueing system. We have proposed two algorithms namely, LORA and p-LORA for resource allocation in loss tolerant multicast video streaming services. We also propose a MWBM algorithm that results in efficient polynomial time implementation of the proposed policies. We propose a modification of the the EXP-Q rule [25] for use in multicast transmission with multiple channels. We perform extensive simulations using video traces from actual video streams [22] to study and compare the performance of LORA, p-LORA and the modified EXP-Q rule. Simulation results indicate that p-LORA results in the least packet loss and the best PSNR of all these policies. Using this policy for
streaming video content in MBMS can significantly reduce resource utilization of video streaming services while also satisfying the video quality requirements of users.

8 Acknowledgement
This work has been supported by the Department of Telecommunications, Ministry of Communications, India as part of the Indigenous 5G Test Bed project.

9 Proofs
9.1 Proof of Theorem 1
We begin by constructing the following randomized resource allocation policy $\Gamma_\delta$ based on $LP(\delta)$ defined in Section 4.2:

Definition 9. Randomized allocation policy $\Gamma_\delta$; $\Gamma_\delta$ chooses an allocation vector in a sub-frame according to a feasible solution $w_C$ of $LP(\delta)$. If the system is in channel state $C$, $\Gamma_\delta$ chooses allocation vector $B_\delta$ w.p. $w_{B,C}$ i.e. $P(B_\delta|\Gamma_\delta|C(t) = C) = w_{B,C}$ \forall $t$ and decisions across sub-frames are independent. $\delta$ is an input parameter for $\Gamma_\delta$.

The definition of $\Gamma_\delta$ will be used for proving various results in this and the following sections. Consider $\lambda \in \Lambda^\circ$. By the definition of $\Lambda^\circ$, this means that, there exists $\delta > 0$ such that $LP(\delta)$ is feasible for arrival rate vector $\lambda$. Let $w_C = \{w_{B,C}\}$ denote a feasible solution of $LP(\delta)$. Therefore, we can use policy $\Gamma_\delta$ to make scheduling decisions in each sub-frame according to $w_C$. Let $A_k[t]$ denote the arrival process of queue $k$. $A_k[t] = 1$ if there is an arrival to queue $k$ in sub-frame $t$ and $0$ otherwise. $D_k^{\Gamma_\delta}[t]$ denotes the departure process of $k$ under $\Gamma_\delta$. $D_k^{\Gamma_\delta}[t] = 1$ if a token departs from $k$ under $\Gamma_\delta$ in sub-frame $t$ and $0$ otherwise.

We have:

$$Q_k^{\Gamma_\delta}[t+1] = \max\{(Q_k^{\Gamma_\delta}[t] + A_k[t] - D_k^{\Gamma_\delta}[t]), 0\},$$

where $Q_k^{\Gamma_\delta}[t]$ is the length of the token queue of UE $k$ at time $t$ under policy $\Gamma_\delta$. For simplicity of notation, we omit the $\delta$ superscript from $Q_k^{\Gamma_\delta}[t]$ and $D_k^{\Gamma_\delta}[t]$ through the rest of this section. Since a departure from queue $k$ means that UE $k$ was successfully served, the corresponding service rate $\mu_k^{\Gamma_\delta}[t] = 1$ and we can write the above equation as:

$$Q_k[t+1] = \max\{(Q_k[t] + A_k[t] - \mu_k^{\Gamma_\delta}[t]), 0\}.$$

The state of the queueing system in a sub-frame can be completely defined by the queue lengths of all the token queues in that sub-frame. We denote the state of the system in sub-frame $t$ by the vector $Q[t] = (Q_1[t], \ldots, Q_M[t])$. Since the scheduling decisions made under $\Gamma_\delta$ only make use of the current state of the system, the evolution of states of the system $\{Q[t]\}_{t \geq 0}$ under $\Gamma_\delta$ forms a Discrete Time Markov Chain (DTMC). This DTMC is countable, irreducible and aperiodic. We prove this in the following result.

Lemma 3. The DTMC $\{Q[t]\}_{t \geq 0}$ is countable, irreducible and aperiodic.

Proof. • Countable: The state space of the DTMC is the set of all $M$-tuples $(Q_1[t], \ldots, Q_M[t])$ where $Q_k[t] \in \mathbb{N}$. It forms an $M$ dimensional Cartesian product of the set of natural numbers $\mathbb{N}$ which is a countable set. Therefore, the state space of the DTMC and hence the DTMC itself is countable (by Theorem 2.13 in [31]).

• Irreducible: The DTMC can transition from any state $Q$ to a state $Q'$ in the following steps:
  - Step 1: Schedule all UEs for service until all queues are empty. This is accomplished in $max_k Q_k$ sub-frames.
  - Step 2: For the next $max_k Q'_k$ sub-frames, the token queue of UE $k$ has an arrival and no departure for the first $Q'_k$ sub-frames. In the remaining $(max_k Q'_k - Q'_k)$ sub-frames, there is no new arrival and no departure. At the end of this step, the DTMC is in state $Q'$.

These steps define at least one path of length $(max_k Q_k + max_k Q'_k)$ from any state $Q$ to any other state $Q'$. Therefore, the DTMC is irreducible.

• Aperiodic: If the DTMC is in state $Q[t]$ and no new token arrives in any queue and no queue is scheduled for service, the state of the DTMC remains unchanged. Therefore, self loops exist and the DTMC is aperiodic.

We now begin the proof of Theorem 1.

Proof. We prove Theorem 1 in two steps. First we establish that $\Lambda^\circ \subseteq S$ in Lemma 4 and then show that $S \subseteq \bar{X}$ in Lemma 5.

Lemma 4. Every $\lambda \in \Lambda^\circ$ is a stabilizable arrival rate vector. Hence, $\Lambda^\circ \subseteq S$.

Proof. To prove this, we first show using Foster’s theorem [32] that DTMC $\{Q[t]\}_{t \geq 0}$ is positive recurrent and hence the queue lengths do not grow infinitely under $\Gamma_\delta$. Using the Lyapunov function $f(Q[t]) = \sum_{k=1}^{M} Q_k[t]$, we have:

$$f(Q[t+1]) - f(Q[t]) \leq \sum_{k=1}^{M} (A_k[t] - \mu_k^{\Gamma_\delta}[t])^2 + 2Q_k[t](A_k[t] - \mu_k^{\Gamma_\delta}[t]).$$

Hence,

$$\mathbb{E}[(f(Q[t+1]) - f(Q[t]))|Q[t]] \leq \mathbb{E} \left[\sum_{k=1}^{M} (A_k[t] - \mu_k^{\Gamma_\delta}[t])^2 + 2Q_k[t](A_k[t] - \mu_k^{\Gamma_\delta}[t])\right]|Q[t]],$$

$$\leq M + 2\mathbb{E} \left[\sum_{k=1}^{M} Q_k[t]A_k[t] - \sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma_\delta}[t]\right]|Q[t],$$

$$\leq M + 2\mathbb{E} \left[\sum_{k=1}^{M} Q_k[t]A_k[t] - \sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma_\delta}[t]\right]|Q[t].$$

From $LP(\delta)$, we have $\mathbb{E} \left[\mu_k^{\Gamma_\delta}[t]|Q[t]\right] = \lambda_k + \delta$. Therefore,

$$\mathbb{E}[(f(Q[t+1]) - f(Q[t]))|Q[t]] \leq M + 2\mathbb{E} \left[\sum_{k=1}^{M} Q_k[t]A_k[t] - \sum_{k=1}^{M} Q_k[t](\lambda_k + \delta),$$

$$\leq M - 2\mathbb{E} \sum_{k=1}^{M} Q_k[t]|.\delta.$$

Defining set $A = \{Q : \sum_{k=1}^{M} Q_k \leq \frac{M+1}{2\delta}\}$, we have:

$$\mathbb{E}[(f(Q[t+1]) - f(Q[t]))|Q[t]] < \begin{cases} -1, & \forall Q[t] \notin A, \\ \infty, & \text{otherwise.} \end{cases}$$

Thus, by Foster’s theorem [32], the DTMC is positive recurrent so the expected queue lengths in the queueing system are finite. Therefore, $\Gamma_\delta$ stabilizes the system for arrival rate vector $\lambda \in \Lambda^\circ$. Thus, $\lambda \in S$ which implies that $\Lambda^\circ \subseteq S$. □
This proves the first part of our result. We now need to show that \( S \subseteq \mathcal{X} \). In the interest of simplicity of notation, we assume that under a policy \( \Gamma \) that stabilizes the system, the following limit exists w.p. 1,

\[
\lim_{t \to \infty} \frac{1}{T} \sum_{i=1}^{T} \mathbf{1}_{B_i,C}[t],
\]

where \( \mathbf{1}_{B_i,C}[t] \) is an indicator random variable that is 1 if allocation vector \( B_i \) is chosen by \( \Gamma \) under channel state \( C \) in sub-frame \( t \) and zero otherwise. Now consider the following sets of sample paths:

- \( A_1 \): the set of sample paths on which Strong Law of Large Numbers (SLLN) holds for the arrival rates i.e. \( \frac{1}{t} \sum_{i=1}^{t} \lambda_i[t] \to \lambda_k \) as \( t \to \infty \), \( \forall \ k \). This is a probability 1 set i.e. \( P(A_1) = 1 \).
- \( A_2 \): set of sample paths on which \( \frac{1}{t} \sum_{i=1}^{t} \lambda_i[t] \to g(C) \) as \( t \to \infty \), \( \forall \ C \) (SLLN holds) where \( \mathbf{1}_{\{C(t) = C\}} \) is an indicator random variable that is 1 if the channel state at sub-frame \( t \) is \( C \) and 0 otherwise. Since \( g \) is a probability distribution over the set of channel states \( C \), we have, \( P(A_2) = 1 \).
- \( A_3 \): the set of sample paths on which service rate under \( \Gamma \) is \( \geq \lambda \). Since \( \Gamma \) stabilizes the system, we have \( P(A_3) = 1 \).
- \( A_4 \): the set of sample paths over which the limit in (8) exists. Since we assume that this limit exists w.p. 1, \( P(A_4) = 1 \).

Since \( A_1, A_2, A_3, A_4 \) are probability 1 sets, their intersection,

\[
A = \bigcap_{i=1}^{4} A_i \tag{9}
\]

is also a probability 1 set. We refer to the sample paths belonging to this set \( A \) as non-trivial sample paths.

We now prove the second part of our result.

**Lemma 5.** If \( \lambda \notin \mathcal{X} \), then \( \lambda \notin S \). Thus, \( S \subseteq \mathcal{X} \).

**Proof.** We prove this result using a contradiction. Let \( \lambda \notin \mathcal{X} \) be a stabilizable arrival rate vector i.e. \( \lambda \in S \). Since \( \lambda \) is a stabilizable arrival rate vector, there exists some allocation policy \( \Gamma \) that can stabilize the system for arrival rate \( \lambda \).

We observe the scheduling decisions made by this \( \Gamma \) along a non-trivial sample path from the set \( A \) defined in (9). Let \( \mathbf{1}_{B_i,C} \) denote the fraction of time for which \( \Gamma \) chooses the allocation vector \( B_i \) in channel state \( C \) along such a sample path. Since \( \Gamma \) stabilizes the system, the rate of departures must equal the arrival rate in the system. Therefore:

\[
\sum_{C \in C} \sum_{B_i \in B} g(C) v_{B,C} \mu_{B,C} = \lambda,
\]

where \( v_{B,C} \geq 0 \) \( \forall B_i \in B, C \in C \),

\[
\sum_{B_i \in B} v_{B,C} = 1 \forall C \in C.
\]

This implies that \( v = \{v_{B,C}\} \) is a feasible solution of LP(\( \delta \)) and that,

\[
\lambda \in \Lambda(0) \implies \lambda \in \mathcal{X}, \tag{10}
\]

which is a contradiction. Therefore, \( \lambda \notin \mathcal{X} \) is not stabilizable i.e. any stabilizable \( \lambda \) must be contained in \( \mathcal{X} \). Hence, \( S \subseteq \mathcal{X} \).

Now, let us assume that the loss requirement of UE \( k \) is met. We show that this ensures the stability of its token queue. Since the loss requirement \( \ell_k \) is achievable, there exists a policy \( \Gamma \) that satisfies the loss requirement. This means that, under \( \Gamma \), the UE is being served in greater than \( (1 - \ell_k) \) fraction of sub-frames. Since the arrival rate \( \lambda_k = (1 - \ell_k) \), the queue is served at a rate greater than the arrival rate. Hence, \( \Gamma \) stabilizes the token queue. From these arguments, we conclude that the loss requirement of a UE is met iff its corresponding token queue in the queueing system is stable. Therefore, the feasible region of the optimal allocation policy \( \Gamma^* \), \( L^{\Gamma^*} \) is equivalent to the stability region of the queueing system, \( S \).

\[\square\]

**9.2 Proof of Theorem 2**

**Proof.** We need to show that the loss requirement of a UE is met iff its token queue in the queueing system is stable. We first argue that the stability of the queueing system implies that the loss requirements of the UEs are met. If the queue corresponding to UE \( k \) is stable, it means that there exists a policy \( \Gamma \) that stabilizes the queue for \( \lambda \in \Lambda^0 \). We can, therefore, construct a randomized policy \( \Gamma_\delta \) as defined in Definition 9 above. Under \( \Gamma_\delta \), the rate of service is greater than \( \lambda_k \) which means that UE \( k \) is served in greater than \( (1 - \ell_k) \) of the sub-frames. Therefore, the loss encountered by UE \( k \) is less than \( \ell_k \) and its loss requirement is met.

Now, let us assume that the loss requirement of UE \( k \) is met. We show that this ensures the stability of its token queue. Since the loss requirement \( \ell_k \) is achievable, there exists a policy \( \Gamma \) that satisfies the loss requirement. This means that, under \( \Gamma \), the UE is being served in greater than \( (1 - \ell_k) \) fraction of sub-frames. Since the arrival rate \( \lambda_k = (1 - \ell_k) \), the queue is served at a rate greater than the arrival rate. Hence, \( \Gamma \) stabilizes the token queue. From these arguments, we conclude that the loss requirement of a UE is met iff its corresponding token queue in the queueing system is stable. Therefore, the feasible region of the optimal allocation policy \( \Gamma^* \), \( L^{\Gamma^*} \) is equivalent to the stability region of the queueing system, \( S \).

\[\square\]

**9.3 Proof of Theorem 3**

**Proof.** Let \( D_k^{\Gamma_0}[t] \) denote the departure process of queue \( k \) under \( \Gamma_0 \). We have:

\[
Q_k^{\Gamma_0}[t + 1] = \max \{ (Q_k^{\Gamma_0}[t] + A_k[t] - D_k^{\Gamma_0}[t], 0) \},
\]

where \( Q_k^{\Gamma_0}[t] \) denotes the queue length of the token queue of \( k \) at time \( t \) under \( \Gamma_0 \). For the sake of simplicity of notation, we omit the \( \Gamma_0 \) superscript from \( Q_k^{\Gamma_0}[t] \) and \( D_k^{\Gamma_0}[t] \) through the rest of this section. Since a departure from queue \( k \) means that UE \( k \) was successfully served, the service rate \( \mu_k^{\Gamma_0}[t] = 1 \) and we can write the above equation as:

\[
Q_k[t + 1] = \max \{ (Q_k[t] + A_k[t] - \mu_k^{\Gamma_0}[t], 0) \}.
\]

The state of the queueing system is completely defined by the vector \( Q[t] = [Q_1[t], \ldots, Q_M[t]] \). The evolution of \( Q[t] \) forms a DTMC since the scheduling decisions made by \( \Gamma_0 \) in a sub-frame are based solely on the state of the system in that sub-frame. The DTMC is countable, irreducible and aperiodic. The proof that the DTMC has these properties follows the same arguments as in Lemma 3 in Section 9.1. We now show using Foster's theorem [32] that this DTMC is positive recurrent and hence the token queues do not grow infinitely.

Using the Lyapunov function \( f(Q[t]) = \sum_{k=1}^{M} Q_k^2[t] \), we have:

\[
f(Q[t + 1]) - f(Q[t]) = \sum_{k=1}^{M} [(A_k[t] - \mu_k^{\Gamma_0}[t])^2 + 2Q_k[t](A_k[t] - \mu_k^{\Gamma_0}[t])].
\]

\[\square\]
Hence,
\[
E[(f(Q[t+1]) - f(Q[t]))|Q[t]] = E\left[\left(\sum_{k=1}^{M} (A_k[t] - \mu_k^{\Gamma}[t])^2 + 2Q_k[t](A_k[t] - \mu_k^{\Gamma}[t])\right)|Q[t]\right] \\
\leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k - 2E\left[\left(\sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma}[t]\right)|Q[t]\right].
\] (11)

Let \(\mu_k^{\Gamma}[t]\) denote the service rate for UE \(k\) in sub-frame \(t\) under the randomized policy \(\Gamma_\delta\). Then, from (2), we have:
\[
\sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma}[t] \geq \sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma}[t].
\] (12)

Therefore, from (11) and (12):
\[
E[(f(Q[t+1]) - f(Q[t]))|Q[t]] \leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k - 2E\left[\left(\sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma}[t]\right)|Q[t]\right], \\
\leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k - 2\sum_{k=1}^{M} Q_k[t](\lambda_k + \delta), \\
\leq M - 2\sum_{k=1}^{M} Q_k[t]\delta.
\]

Now for set \(A = \{Q : \sum_{k=1}^{M} Q_k \leq \frac{M+1}{2}\}\), we have:
\[
E[(f(Q[t+1]) - f(Q[t]))|Q[t]] < \left\{ \begin{array}{ll}
-1, & \forall Q[t] \notin A, \\
\infty, & \text{otherwise}.
\end{array} \right.
\]

Thus, by Foster’s theorem [32], the DTMC is positive recurrent which means that the expected queue lengths in the queueing system will be finite. Therefore, \(\Gamma_0\) stabilizes the system and hence meets the loss requirements of the UEs. \(\blacksquare\)

### 9.4 Proof of Lemma 1

For the sake of simplicity of notation, we omit the \(\Gamma_P\) superscript from the notations through the rest of this section.

**Proof.**

- **Countable:** The state of the DTMC \(Q[t]\) comprises the queue lengths of \(M\) UEs and their priority weights. We have already shown in Lemma 3 that the state space of queue lengths \(\{Q_1[t], \ldots, Q_M[t]\}\) is a countable set. The state space of the priority weights of the UEs is an \(M\) dimensional Cartesian product over the finite set \(\{1, 2, \ldots, \kappa\}\) and is therefore a finite countable set (Theorem 2.13 in [31]). Therefore, the states of the DTMC \(Q[t]\) form a \(2M\) dimensional Cartesian product over two countable sets, the state space of queue lengths and the state space of the priority weights. Therefore, the state space of the DTMC and hence the DTMC itself is countable (Theorem 2.13 of [31]).

- **Irreducible:** Consider that the DTMC is in state \(Q = \{Q_1, \ldots, Q_M, \bar{c}_k\}\). We will show that a path exists from \(Q\) to any state \(Q' = \{Q'_1, \ldots, Q'_M, \bar{c}'_k\}\). The DTMC can transition from \(Q\) to \(Q'\) in the following steps:

  - Step 1: Schedule all UEs for service until all queues are empty. This is accomplished in \(\max_k Q_k\) sub-frames.
  - Step 2: A new token arrives in every queue and no queue is scheduled for service for the next \(\min_k Q_k'\) sub-frames. At the end of this step, all queue lengths are equal to \(\min_k Q_k'\) and all priority weights are equal to \(\min(\min_k Q_k', \kappa)\).

  - Step 3: For the next \(\max_k Q_k' - \min_k Q_k'\) sub-frames, the UEs in \(\arg \max_k Q_k'\) see one arrival and no departure.

  - Step 4: Every other UE \(k'\) see an arrival and no departure for the first \(\max_k Q_k' - \min_k Q_k'\) sub-frames and one arrival and one departure for the remaining \(\max_k Q_k' - Q_{k'}\) sub-frames. At the end of this step, the queue length of UE \(k\) is equal to \(Q_k\).

  - Step 5: In the next \(\max_k c_k'\) sub-frames, there is no arrival and no departure for UEs in \(\arg \max_k c_k'\). For every other UE \(k'\), there is an arrival and a departure in the first \(\max_k c_k' - c_k'\) of these sub-frames and no arrival and no departure in the remaining \(\max_k c_k'\) sub-frames. At the end of this step, the DTMC is in the desired state \(Q'\).

This defines one finite length path from any state \(Q\) to any other state \(Q'\) of length \((\max_k Q_k + \max_k Q_k' + 1 + \max_k c_k')\). Hence, the DTMC is irreducible.

- **Aperiodic:** Consider state \(Q[t]\) where all queues are empty and all priority weights are 0. If there is one arrival in each queue in slot \(t + 1\) and every queue is served, the queues remain empty and the priority weights remain 0. Therefore, this state has a self loop and hence has period 1. Since we have already shown that the DTMC is irreducible, all states have period 1 because periodicity is a class property. Hence, the DTMC is aperiodic. \(\blacksquare\)

### 9.5 Proof of Theorem 4

**Proof.** Let \(D_k^{\Gamma}[t]\) denote the departure process of queue \(k\) under \(\Gamma_P\). We have:
\[
Q_k^{\Gamma}[t + 1] = \max\{(Q_k^{\Gamma}[t] + A_k[t] - D_k^{\Gamma}[t]), 0\},
\]
where \(Q_k^{\Gamma}[t]\) is the queue length of queue \(k\) at time \(t\) under \(\Gamma_P\). For simplicity of notation, we omit the \(\Gamma_P\) superscript from \(Q_k^{\Gamma}[t]\) and \(D_k^{\Gamma}[t]\) through the rest of this section. Since a departure from queue \(k\) in sub-frame \(t\) means that \(\mu_k^{\Gamma}[t] = 1\), we can write the above equation as:
\[
Q_k[t + 1] = \max\{(Q_k[t] + A_k[t] - \mu_k^{\Gamma}[t]), 0\}.
\]

Under this policy, the evolution of the state of the queueing system \(Q[t] = [Q_1[t], \ldots, Q_M[t], \bar{c}[t]]\) forms a DTMC. We have proved in Lemma 1 that this DTMC is countable, irreducible and aperiodic. We now show using Foster’s theorem [32] that this DTMC is positive recurrent and hence the queues do not grow infinitely.

Using the following Lyapunov function \(f(Q[t]) = \sum_{k=1}^{M} Q_k^2[t]\), we have:
\[
f(Q[t+1]) - f(Q[t]) = \sum_{k=1}^{M} [(A_k[t] - \mu_k^{\Gamma}[t])^2 + 2Q_k[t](A_k[t] - \mu_k^{\Gamma}[t])].
\]

Hence, as in (11), we have:
\[
E[(f(Q[t+1]) - f(Q[t]))|Q[t]] \leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k - 2E\left[\left(\sum_{k=1}^{M} Q_k[t]\mu_k^{\Gamma}[t]\right)|Q[t]\right].
\] (13)
Let $\mu_k^{G_k}[t]$ denote the service rate for UE $k$ in sub-frame $t$ under the randomized policy $\Gamma_0$. Then, from (3), we have:

$$\sum_{k=1}^{M} \left( Q_k[t] \mu_k^{G_k}[t] + (ck[t]+1)sp_k^{G_k}[t] \right)$$

$$\geq \sum_{k=1}^{M} \left( Q_k[t] \mu_k^{*}[t] + (ck[t]+1)sp_k^{*}[t] \right).$$  (14)

Therefore, from (13) and (14):

$$\mathbb{E}(f(Q[t+1]) - f(Q[t]))|Q[t] \leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k$$

$$- 2E \left[ \sum_{k=1}^{M} Q_k[t] \mu_k^{G_k}[t] + (ck[t]+1)(\mu_k^{G_k}[t] - \mu_k^{*}[t])s \right] Q[t] \right),$$

$$\leq M + 2\sum_{k=1}^{M} Q_k[t]\lambda_k - 2\sum_{k=1}^{M} Q_k[t] (\lambda_k + \delta)$$

$$- 2E \left[ \sum_{k=1}^{M} - (k+1)s \right],$$

$$\leq M - 2\sum_{k=1}^{M} Q_k[t]\delta + 4Ms. \text{ (for } \kappa = 1 \text{)}$$

Defining set $A = \{ Q : \sum_{k=1}^{M} Q_k[t] \leq \frac{MS + M + 1}{2d} \}$, we have:

$$\mathbb{E}(f(Q[t+1]) - f(Q[t]))|Q[t] < \left\{ \begin{array}{ll} -1, & Q[t] \not\in A, \\ \infty, & \text{otherwise}. \end{array} \right.$$  

Thus, by Foster's theorem [32], the DTMC is positive recurrent meaning that the expected queue lengths in the queuing system will be finite. So, $\Gamma_0$ stabilizes the system and hence meets the loss requirements of all the UEs. $\Box$

9.6 Proof of Lemma 2

Proof. A matching for graph $G$ selects edges that share no common vertices. This means that each group from $U$ will be matched to exactly one PRB from $V$ and each PRB from $V$ will be matched to at most one group from $U$. Therefore, the requirement of assigning no more than 1 PRB to each group is satisfied. Since PRBs in $V$ are matched to no more than one group from $U$, we will have $B_i^k[t] \neq B_i'^k[t] \forall (i, i') \in [L] : B_i^k[t], B_i'^k[t] \neq 0$ as required by Definition 1. Thus, the solution of the MWBM gives us a feasible resource allocation. Next, we show that the resulting allocation is consistent with the allocation decisions that would be made by policy $\Gamma_0$.

MWBM picks edges such that the sum of the weights of the edges chosen is maximized. Therefore, it maximizes the quantity $\sum_{i \in U} \sum_{k \in G_i} Q_k[t]v_k[i] = \sum_{k=1}^{M} Q_k[t] \mu_k^{*}[t]$ which is same as in (2). Hence, resource allocation done using MWBM on $G$ is consistent with policy $\Gamma_0$. $\Box$

References