OPTIMAL QAM ASSIGNMENT IN THE PRESENCE OF MIXED SD AND HD STREAMS

Jiong Gong andYasser Syed
Cable Television Laboratories, Inc.

Abstract

When VoD systems need to accommodate both SD and HD streams, the traditional capacity engineering rule for deploying QAM modulators face a new challenge. Two issues arise from this new paradigm. One is the streaming bit rate for HD, and perhaps both HD and SD streams to optimize the system performance. The second issue is the QAM allocation algorithm to minimize system inefficiencies. We propose a solution that has the potential to significantly improve system performance to accommodate a mixture of SD and HD VoD streams compared to prevalent methods.

INTRODUCTION

With the rapid penetration of HD TV sets in the consumer electronics market, cable companies have been aggressively deploying High Definition (HD) cable channels in recent years. A powerful competitive response to the DBS offerings is the HD VoD service. When the VoD streams consist of only one type that is of Standard Definition (SD) TV, the Quadrature Amplitude Modulation (QAM) assignment algorithm is not critical in affecting overall system performance. Commonly implemented algorithms include starting from the busiest QAM and starting from the least-busy QAM. Either way is not going to affect the blocking rate of the system. To make full use of a QAM channel capacity, bit rate is often chosen so that all possible numbers of streams in a channel form a congruence class of a modulo in the streaming bit rate with the division remainder to be zero.

When VoD systems need to accommodate both SD and HD streams, the traditional stream capacity assignment rule for QAM modulators faces a new challenge, in that it is possible to incur blockage on each of the QAM modulators while jointly they have the capacity to support the arrival of a new stream request. In other words, the posterior allocation of the streams is suboptimal. Both the busiest and the least-busy rule tend to have suboptimal allocations. This brings about the issue of finding an alternative algorithm to improve the allocation efficiency. We propose in this paper an optimal solution that is a significant improvement over current methods.

VOD STREAMING BIT RATES

Before getting into QAM allocation algorithms, it is important to first look at the issue of VOD streaming bit rates. Design of bit rates should consider both the issue of fully using QAM resources and the quality issue perceived by viewers. For each type of stream to fully utilize the useable bandwidth capacity of a QAM, there must be a modular relationship between each type of stream. Additionally, if a different capacity QAM is used then a modular relationship must still also exist for full utilization. In reality, the limits of this relationship are dependent on the modulation types in use. Since quality is dependent on bit rate, there is an additional limit to designing the modularity factor between the two types of streams.

There are some real practical limitations on this relationship. The data rate used in a QAM-designated VOD service today is 37.5 Mbps for 256 QAM and 26.25 Mbps for 64 QAM. The additional capacity in the data rate is reserved for in-band traffic. Today a typical SD VOD stream is at a constant bit rate of 3.75 Mbps which is good quality for MPEG-2 paid movie content. If one wanted to determine the HD MPEG-2 stream bit rate based on modularity on a fully-utilized QAM, then HD date rates would be 7.5 Mbps (2x of SD stream) or 18.75 Mbps (5x of SD stream), which is either too low in quality or too much data rate used. Alternatives to this would be 11.25 Mbps (3x of SD stream) or 15 Mbps (4x of SD stream). Both of these offer acceptable quality, but they are not fully-useable "pure" QAMs, requiring either a 2 HD/2 SD- 256 QAM (or 1 HD/2 SD- 64 QAM) stream configuration or 3 HD stream/1 SD stream-256 QAM (or 2HD/1SD-64 QAM) configuration because of a modular relationship.

In typical systems, a VOD service to a node is allocated in 4 QAMs (for now let's assume 256 QAM) or an integer multiple of it. This has to do basically with fiber distribution to a node. For a pure SD VOD service, this would be about 40 streams that could service about 400 homes, assuming a 10% peak capacity. For a pure HD VOD service, this could be either 8 to 12 streams that could service from 80 to 120 homes, assuming a 10 % peak capacity. If a QAM allocation algorithm is not properly configured, this would either lead to blocking of HD streams to 4 to 8 streams by the wrong placement of 4 SD streams. That leads to only 40 to 80 households for HD VoD service, assuing 10% peak usage rate.

The amount of time of this blocking would depend on length of overlap that the bandwidth is reserved for each movie. The current pratice in many VoD systems is to tear down the stream, if the stream incurs more than 5 minutes of pausing. One futher complexity is whether the torn down stream needs to have any priority over new stream requests, if it needs to be resumed again. If so, this would affect how the bandwidth is allocated and the amount of time the bandwidth is reserved (e.g. a 2 hour movie may typically have a reserve time that has an extra 20-30 minutes).

QAM RESOURCE ALLOCATION

In this section, we describe a QAM allocation algorithm that we believe represents a significant improvement over current prevalent methods. We start off by describing a mathematical framework to model the problem. Suppose a collection of *n* QAM modulators is deployed to serve a VoD service group. Let q_i, $i = 1, 2, ..., n$, denote the used capacity of each QAM modulator *i*. Total capacity, *Q*, is assumed to be the same for all QAM modulators. Therefore, the remaining capacity that can be used for new stream requests on that QAM modulator is then $Q - q_i$. Let r_s and r_h denote the streaming bit rate respectively for SD and HD streams. The two types of streams may arrive at a collection of QAM resources according to two distinct random processes, such as the Poisson process, but exit the system based on the same holding time distribution. We call the snap shot state of all QAMs $[q_i]$ at a particular time as an allocation. We define an allocation as inefficient, if,

$$Q - q_i < r_h, \ \forall i, \text{ and } \sum Q - q_i \geq r_h$$

In other words, none of the QAMs individually has the capacity, even though the sum of all available resources on each QAM is able to support a HD stream request.

Note that while each type of stream is in itself modulus in its own bit rate, they jointly are not when they are mixed together in a QAM. As a result, inefficiency tends to arise when streams are mixed together. Both busiest and least-busy algorithms tend to create mixing QAMs (i.e., both SD and HD streams carried by the same QAM) as the joint process of the two stream types is a mixture of two random processes. Naturally our improvement over the current methods is in the direction of minimizing the probability of a mixing QAM.

The state of each QAM modulator can be categorized into four possible types:

- It is entirely empty.
- A mixture of SD and HD streams are occupying it.
- Only SD streams are occupying it.
- Only HD streams are occupying it.

Mathematically we denote these four types accordingly by defining a state function as:

$$S_i(q_i) = \begin{cases} 1, \ if \ q_i = 0 \\ 2, \ if \ q_i = x_i r_s + y_i r_h, \ x_i \neq 0, \ y_i \neq 0 \\ 3, \ if \ q_i = x_i r_h, \ x_i \neq 0 \\ 4, \ \ if \ q_i = y_i r_h, \ y_i \neq 0 \end{cases}$$

where x_i and y_i are positive integers representing the number of SD and HD streams occupying QAM modulator *i*. In the above four states, we call a QAM in state 1 an empty QAM. We call a QAM in state 2, that is $S_i(q_i) = 2$, a mixing QAM. QAMs in state 3 and 4 are called non-mixing SD and HD QAMs respectively.

Since our algorithm is based on the principle of minimizing the probability of mixing two types of streams within a QAM modulator, it is then straightforward to prioritize over the states of available QAM modulators for stream allocation. The first priority is to go with a non-mixing QAM of the same stream type. The next priority is to go with an empty QAM. Then go with a mixing QAM. The last resort is to create another mixing QAM – going with a non-mixing QAM of the opposite type.

If there are multiple QAM modulators available within the same state class, priority is given to those QAM modulators that have a larger likelihood of becoming a non-mixing QAM or an empty QAM once some streams start to drop. This implies the following rules:

If multiple non-mixing QAMs are available to a stream request of the same stream type, priority should be given to the busiest non-mixing QAM, because other non-mixing QAMs have a higher likelihood of being empty.

If multiple mixing QAMs are available to a SD or HD stream request, priority is given to the busiest mixing QAM, because other mixing QAMs have a higher likelihood of being non-mixing or empty.

If multiple non-mixing QAMs are available to a stream request of a different type, that is if a stream request will have to create a new mixing QAM, priority is given to the least busy QAM, because it has the highest likelihood of becoming non-mixing again.

With these rules explained, the selection algorithm of a particular QAM modulator by a new stream request is then based on the following sequence. We take a SD stream request as an example.

1. Identify a set of I, s.t. $Q - q_i \geq r_s$ for $\forall i,\ i \in I$

1.1 If I is empty, reject the stream request;

2. Identify a subset of J, $J \subseteq I$, s.t. $S_j(q_j) = 3,\ j \in J$;

2.1 If J is empty, go to the next step;

2.2 If J has multiple elements, select $j^* = \arg \underset{j \in J}{Min}\ Q\text{-}q_j$;

2.3 If there are multiple j^*, select randomly among j^*;

3. Identify a subset of J, $J \subseteq I$, s.t. $S_j(q_j) = 1,\ j \in J$;

3.1 If J is empty, go to the next step;

3.2 If J has multiple elements, select j^* randomly;

4. Identify a subset of J, $J \subseteq I$, s.t. $S_j(q_j) = 2,\ j \in J$;

4.1 If J is empty, go to the next step;

4.2 If J has multiple elements, select $j^* = \arg \underset{j \in J}{Min}\ Q\text{-}q_j$

4.3 If there are multiple j^*, select randomly among j^*;

5. Identify a subset of J, $J \subseteq I$, s.t. $S_j(q_j) = 4,\ j \in J$;

5.1 If J has multiple elements, select $j^* = \arg \underset{j \in J}{Max}\ Q\text{-}q_j$;

5.2 If there are multiple j^*, select randomly among j^*;

We conclude this paper by presenting the major result of the paper as in the following proposition. The result waits to be verified by simulation tests. The test scheme will generate two arrival random processes and document blocking rates for the three alternative algorithms. If our algorithm generates least blocking under the same load, it would then verify the theoretical result.

Proposition: The QAM allocation algorithm described above is more efficient than the busiest and the least-busy algorithms.

Proof: Suppose the system has only empty and non-mixing QAMs. At this point, the system has capacity to accommodate new streams. As more streams are added to the system, the algorithm can only incur mixing when the last QAM is called for to meet the demand. In other words, mixing occurs when the system is close to full capacity except the last QAM. Note at this point, all other QAMs are non-mixing. Let j denote the only mixing QAM. When it is full, $Q - q_j < r_h$ implies $\sum Q - q_i = Q - q_j < r_h$. This is because all other QAMs are non-mixing, and each type of stream is modulus in its own bit rate. Therefore there would be no inefficiency according to our definition. Therefore our algorithm is more efficient.

On the other hand, mixing of QAM in the busiest and the least-busy algorithms is a random event. Therefore inefficiency is likely to result with a higher probability.

CONCLUSIONS

This paper presents an alternative QAM resource allocation algorithm to accommodate a mixture of SD and HD VOD streams. Our analysis indicates the need to first design the stream bit rates so as to be modular with respect to the full QAM capacity, as well as to be modular with respect to each other. The subsequent allocation algorithm then calls for avoiding the mixing of different types of streams to the extent possible. This principle is very much like the principle of ocean freight container shipment, where modularity is critical in making full and efficient use of the ship capacity.

In the future when there are more types of streams in the same QAM set (e.g., alternate codec versions like MPEG 4-AVC or VC-1, and each with an SD or HD version), one should consider the modularity for selecting the bit rates for these. Additionally if we consider supporting VBR streams instead of CBR streams. Some parameters may need to reflect modularity, start time difference in QAM streams, and value of the streams.