DOCSIS 3.1 Profile Management Application and Algorithms

Greg White and Karthik Sundaresan
Cable Television Laboratories, Inc. (CableLabs)

Abstract

DOCSIS 3.1 OFDM Profiles provide a wide range of modulation choices that can be used to fine-tune the CMTS's transmissions to get the best performance from the current network conditions. A well-designed, optimized set of modulation profiles allows a downstream channel to operate with a lower SNR margin, allowing a channel to operate at an overall higher throughput.

This paper describes methods for designing OFDM profiles and choosing the appropriate modulation orders for a profile. It answers the questions around which profile is appropriate for a CM and what is the optimal set of profiles to use across the an OFDM channel for a given set of CMs. This paper defines an objective function which can be used to calculate the gain in system capacity resulting from the use of multiple profiles, and then explores approaches to maximizing that objective function for a population of CMs. It proposes one such approach, which is referred to as the K-means Coalescation Algorithm, that appears to provide very good results with low computational complexity.

INTRODUCTION

The DOCSIS 3.1 (D3.1) specification fundamentally changes the nature of information delivery across the cable plant, and along with that, the way HFC networks will be maintained and managed. For the first time in the history of HFC, the downstream channel does not use a one-size-fits-all modulation scheme; rather, the modulation and forward error correction can be optimized based on actual plant conditions at individual devices, so that devices that receive an exceptionally clean signal can utilize very efficient high-order-modulation, while devices that have a degraded signal will use more robust modulation, all on the same channel.

The DOCSIS 3.1 toolbox provides a wide range of modulation choices that can be used to fine-tune the transmissions to get the best performance from the current network conditions. To manage the optimization of these settings across the population of devices, the CMTS uses the concept of downstream profiles. A downstream profile defines the modulation order (i.e., bit loading) on each of the 3840 or 7680 subcarriers across the OFDM channel. D3.1 provides for defining multiple downstream profiles, where each profile can be tuned to account for the specific plant conditions that are experienced by a set of CMs.

A well-designed, optimized set of modulation profiles allows a downstream channel to operate with a lower SNR margin, potentially allowing a channel to operate at an overall higher throughput. In addition, it can account for troubled devices by providing service even in situations where significant plant impairments exist.

The application that implements this optimization logic can be external to a CMTS, enabling the most efficient use of profiles across channels and CMs. For an operator, it also allows uniform operation of such algorithms across different CMTS platforms. This profile optimization and profile creation functionality is implemented as an 'application' running outside the CMTS and is known in this paper as the Profile Management Application (PMA).

This paper briefly describes the data elements needed to enable a Profile Management Application, the key information needed to make decisions (e.g., channel config, profile test results, SNR/MER data for each CM etc.). It then focuses on the PMA itself, and the algorithms it needs to implement. This paper analyzes how a PMA can choose the appropriate modulation orders for a profile and other related questions such as: How does the PMA decide which profile is appropriate for a CM? How does the PMA decide the optimal set of profiles to use across the network?

DOCSIS DOWNSTREAM OFDM CHANNELS

A D3.1 CM supports two or more independently configurable OFDM channels, each occupying a spectrum of up to 192 MHz in the downstream. The maximum channel bandwidth of 192 MHz corresponds to 3841 subcarriers in 4K mode and 7681 subcarriers in 8K mode. The OFDM signal is composed of: Data subcarriers, Scattered pilots, Continuous pilots and PLC subcarriers. A DOCSIS 3.1 downstream OFDM 192 MHz channel's encompassed spectrum (active bandwidth of the channel) does not exceed 190 MHz. So the number of contiguous active subcarriers in a downstream OFDM channel does not exceed 3800 for 4K FFT and 7600 for 8K FFT. The two downstream channel types are summarized in Table 1.

Table 1 – Downstream Channel Options

Subcarrier Spacing	25 kHz	50 KHz
Symbol Period	40 µs	20 µs
FFT Size	8192	4096
Maximum Number of subcarriers	7600	3800

The downstream OFDM channel bandwidth can vary from 24 MHz to 192 MHz. Bandwidths smaller than 192 MHz are achieved by zero-valuing the subcarriers prior to the IDFT, i.e., by adjusting the equivalent number of active subcarriers while maintaining the same subcarrier spacing of 25 kHz or 50 kHz. See the DOCSIS 3.1 PHY Specification [3] for details on the OFDM Channel definition.

Downstream Modulation Profiles

Each of the subcarriers in the downstream OFDM channel can be configured to use a different modulation order. This allows the CMTS to optimize the downstream transmissions across the wide frequency band (192 MHz) of the channel. The specific choice of modulation order selected for each subcarrier is communicated to the CMs in the form of a modulation profile, which allows them to interpret and demodulate the signal.

A modulation profile consists of a vector of bit-loading values, an integer value for each active subcarrier in the downstream channel. Since the modulation orders range from 16-QAM to 16384-QAM, the range of bit-loading values is from 4 to 14 (skipping 5); however, it is expected that very low bit-loading values, 7 or less, will be used very infrequently since most plants support 256 QAM today.

The CMTS generates a "Profile A" that is the lowest common denominator profile, able to be successfully received by all CMs in the Service Group. It can then generate up to 15 additional modulation profiles, which are communicated to the Service Group. Each CM can be assigned up to four modulation profiles, including Profile A (used for broadcast frames), an optimized profile for the CM's unicast traffic, and possibly two additional profiles that could be used for multicast traffic. Since the number of CMs in the Service Group is expected to be larger than 15 in the majority of cases, each profile is expected to be used by a group of CMs that have similar channel characteristics.

This capability, the ability to optimize the downstream transmission for the channel characteristics of the CM population, is a powerful feature that allows for a significant improvement in channel capacity, and at the same time fundamentally changes the nature of RF plant maintenance. Prior to D3.1, all traffic has been sent using the lowest common denominator modulation (64-QAM or 256-QAM), setting the downstream channel capacity to a fixed value, and setting a Modulation Error Rate (MER) target for plant maintenance. Since only a fraction of the downstream traffic in D3.1 will be carried using Profile A, and since the CMTS can automatically determine the modulation to use for that profile, there is no longer a single MER target. Furthermore, since CMs can be assigned to modulation profiles that are optimized for their channel conditions, there is no longer a fixed value for channel capacity. The cleaner the channel is to any CM, the more efficient its traffic becomes, raising the overall average efficiency, and hence capacity, of the channel. Capitalizing on this capability requires that the CMTS can determine the channel conditions to the set of CMs in the Service Group, and that from this information determine the optimal set of (up to 16) modulation profiles.

OBTAINING INFORMATION FROM THE CMTS

There are a couple of methods which a Profile Management Application can become aware of the channel conditions in the DOCSIS network. These are using the OFDM Downstream Profile Test Request/Response mechanism (OPT-REQ/OPT-RSP MAC Management Message exchanges) or using the Proactive Network Maintenance (PNM) toolset.

OFDM Downstream Profile Test

In a D3.1 network, the CMTS can use new MAC Management Messages like the OPT-REQ to ask a CM to test various aspects of an OFDM downstream channel. A OPT-REQ message to a CM can check the CM's ability to receive a specified downstream OFDM profile and/or query the CM's RxMER statistics.

When asked to report the modulation error ratio measurements, the CM includes the RxMER per Subcarrier in the OPT-RSP message back to the CMTS. These are encoded as a packed sequence of 8-bit values for N consecutive subcarriers ($N \leq 7680$) from lowest active subcarrier to the highest active subcarrier, including all the subcarriers in between. Note that while the vector includes values for excluded subcarriers, a PMA can ignore these values based on its knowledge of excluded subcarriers.

In addition, if needed, a CM can compare the RxMER per Subcarrier to a Threshold and report a result, calculate the number of subcarriers whose RxMER is a certain value below a target, or report the SNR Margin of a candidate profile. See the D3.1 MULPIv3.1 spec for details [4]. In essence a CM can help by precomputing some of the data before sending it back in a OPT-RSP.

When requested to test a candidate profile, a CM reports back the number of Codewords received during testing, Corrected Codeword Count (codewords that failed pre-decoding LDPC syndrome check and passed BCH decoding) and Uncorrectable Codeword Count (failed BCH decoding). In addition, the CM can report if the number of codeword failures were greater than a given threshold for the Candidate Profile.

Proactive Network Maintenance (PNM)

The CMTS and CM perform measurements and report network conditions as a part of supporting PNM functionality in the DOCSIS network. The DOCSIS 3.1 Downstream PNM Measurements and Data include: Symbol Capture, Wideband Spectrum Analysis, Noise Power Ratio Measurement, Channel Estimate Coefficients, Constellation Display, Receive Modulation Error Ratio (RxMER) Per Subcarrier, FEC Statistics, Histogram, and Received Power. See the DOCSIS 3.1 PHY Specification [3] for details.

DOCSIS 3.1 PNM capability assumes a PNM server that initiates PNM tests and receives data output from the CM and/or from the CCAP. Some of the PNM data, like the RxMER numbers, are the same as that computed by a CM for the OPT-RSP. All of this PNM data gets uploaded to a PNM Server When a Profile Management Application (PMA) comes around looking for this data from a PNM server, this data may be dated as compared to obtaining the data using the OPT method.

In this paper we are mainly using the RxMER per sub carrier data from the CMs. The other data as described above are definitely useful and how they can help further refine the profile definition will be a topic for future study.

PROBLEM STATEMENT

Objective of Profile Creation

We assume that a user's demand for bandwidth is independent of the modulation profile to which they are assigned. This assumption is based on the observation that the capability of a CM to use a particular profile is determined by the RF channel characteristics, which are presumably independent of the user themselves. Furthermore, in the absence of a priori information to the contrary, we can assume that, in aggregate, the users assigned to each profile are equivalent in terms of bandwidth usage, and, as a result, each user is expected to place an equal load on the channel on average. (In the case where all profiles contain a significant number of CMs, the law of large numbers would enforce this assumption, but in other cases there may be some achievable gain by taking into account historical usage patterns or weighting users based on their service tier. This is left for future study.) Based on this assumption, we can derive the average channel capacity for a population of CMs divided among a set of modulation profiles as follows.

Each user places data on the channel at a rate of b bits-per-second, and thus each profile of users places data on the channel at a rate of N_x*b bits-per-second, where N_x is the number of users assigned to profile x. This data rate equates to a symbol rate for the profile x, $S_x = N_x b/K_x$ symbols-per-second, where K_x is the total efficiency of profile x (i.e., the sum of the bit-loading values of all of the subcarriers). The total channel symbol rate S is the sum of the symbol rates of all of the profiles (note, this calculation disregards FEC, and the overhead incurred by the use of multiple profiles, i.e., additional NCP blocks, partial codewords, etc.), or:

$$S = \sum_{\forall x} S_x$$

$$S = b \cdot \sum_{\forall x} \frac{N_x}{K_x}$$

Since the symbol rate of the channel is a parameter set by the operator (either 50 Ksym/s or 25 Ksym/s), we can thus derive the per user bit rate for a fully loaded channel as:

$$b = S \Big/ \sum_{\forall x} \frac{N_x}{K_x}$$

And if we calculate the total channel capacity as:

C = N * b, where N is the total number of users.

We can then express this as:

$$C = NS \Big/ \sum_{\forall x} \frac{N_x}{K_x}$$

Or

$$C = S \Big/ \sum_{\forall x} \frac{\Phi_x}{K_x}$$

where $\Phi_x = N_x/N$ is the fraction of users assigned to profile x.

This is the harmonic mean (across all CMs) of each CM's single-user channel capacity.

DOCSIS 3.1 includes the notion of a lowest-common-denominator profile, referred to as "profile A" that can be utilized by all CMs in the service group. It is mandatory that profile A be created (for broadcast data, if nothing else), so a useful metric to assess the utility of a set of candidate profiles P is the ratio of the channel capacity using the set of candidate profiles to the channel capacity only using profile A.

This metric J is thus:

$$J_{P,A} = C_P / C_A$$

$$J_{P,A} = \frac{S / \sum_{\forall x} \frac{\Phi_x}{K_x}}{S / \frac{1}{K_A}}$$

Which reduces to:

$$J_{P,A} = \frac{1}{K_A \cdot \sum_{\forall x \in P} \frac{\Phi_x}{K_x}}$$

And, the objective of profile creation is to select the set of profiles P that maximizes the metric $J_{P,A}$ for the Service Group.

Channel Conditions for the Service Group

Given the population of N CMs in the Service Group, the PMA can generate the N-by-N_sc matrix M describing the bit-loading capability of each CM, where N_sc is the number of active subcarriers in the channel (up to 3800 for the 4K FFT, and up to 7600 for the 8K FFT). The PMA generates this matrix by instructing CMTS/CMs to perform channel measurements and report the MER per subcarrier, and then from this MER, calculating the maximum bit-loading that the CM can reliably receive for each subcarrier. This calculation could include some headroom to ensure reliable communication, or could be aggressive given that FEC will correct for some MER deficiencies. (The DOCSIS 3.1 PHY specification [3] discusses an example where alternating subcarriers are assigned alternating bit-loading values, in order to target a CM that reports an MER that is halfway between the MER required for each of the two bit-loading values, and relying on FEC to enable reliable communication.) The result is a "maximum-bit-loading" vector for each CM.

Calculating Profile A and Φ_x

For a given M matrix, Profile A can be determined by taking the column-wise minimum of M. Furthermore, the values Φ_x are determined by comparing the rows of M to the matrix of profiles P as follows. When considering whether a particular profile is suitable for a CM, the PMA needs to compare the maximum-bit-loading vector for the CM to the bit-loading vector of the profile, and ensure that the CM's vector is greater than the profile vector for all subcarriers. (A more sophisticated algorithm could allow the CM to be somewhat deficient in some number of subcarriers, based on an analysis of the ability for FEC to correct for these deficiencies.) We assume that the PMA will assign each CM to the most efficient profile available, the one with the highest total bit-loading.

This allows us to recast the objective function as:

$J_{P,M} = J_{P,A}$ where A=min(M) and Φ_x is determined as described above.

We can now express the optimization problem more explicitly as: given M, what is the P that maximizes $J_{P,M}$?

M is expected to vary over time, both due to CMs being added to or removed from the Service Group, and due to fluctuations in channel conditions, so it is expected that the PMA would periodically refresh its measurement of M, and update the set of profiles P accordingly.

Profile Cost

As mentioned previously, our derivation for channel capacity, and hence our objective function $J_{P,M}$, doesn't take into account the additional overhead incurred by introducing profiles to the profile set. This "cost" of supporting each additional profile is the result of the additional Next Codeword Pointer Data Message Blocks (needed when the profile in use is changed within a symbol), the possibility of needing to pad a symbol with Zero Bit Loading, and the increased likelihood of using shortened FEC codewords.

Given that $J_{P,M}$ doesn't factor in the "cost" of supporting multiple profiles, it can be shown that the optimal $J_{P,M}$ will increase as the number of profiles in P increases - until the number of profiles equals the number of unique vectors in M (i.e., each CM getting its own personalized profile). So, unless this cost is included in the optimization, the result will be biased toward larger sets of profiles. We leave the analysis of the profile cost for future study, and here focus on the problem of defining/selecting the optimal set of profiles given a target number of profiles, rather than determining the optimal number of profiles.

Multicast Profiles

As mentioned above, some profiles may be used for multicast transmissions that need to be received by a known subset of CMs. The CMTS can very simply generate a multicast profile specifically to target the joiners of each multicast group; however, since the CMTS only supports 16 total profiles, each of these multicast profiles reduces the number of unicast profiles available, and hence reduces the total unicast channel capacity. We leave for future work the problem of optimally defining profiles in the presence of multicast groups, and here focus only on optimizing the unicast profiles.

OPTIMIZATION METHODS

Brute Force Search

One approach to selecting the optimal set of profiles could be a brute force search: try all combinations of all possible profiles, calculating the $J_{P,M}$ value for each combination, and then selecting the best. The difficulty is that the search space is enormous. In a 192 MHz /4K FFT channel, there could be 3800 subcarriers, with 10 choices of bit-loading for each. Thus there are 10^{3800} possible individual profiles, and 10^{3800}-choose-15 possible 16-profile sets (approximately 10^{57000}) to evaluate. In an 8K mode channel (with 7600 subcarriers), the problem is even worse.

The search space can be reduced by observing that there is no need to try profiles that have a lower bit-loading than Profile A. For example, suppose that the geometric mean of the subcarrier bit-loading values of Profile A is 8 (corresponding to 256-QAM), the search space (in the 4K case) is then reduced to 7^{57000} or $\sim 10^{48000}$. This is a significant reduction, but still well beyond being computationally feasible.

Another simplification could be to observe that it is unlikely that subcarrier bit-loading would need to vary by more than one modulation order between adjacent subcarriers. This simplification could bring us down to approximately 10^{27000} profile sets. A further simplification (that may reduce the optimality) would be to assume that bit-loading would only change every 2 MHz (40 subcarriers), and then only by at most one modulation order each time. This reduces the number of profile sets to approx. 10^{700}. This is a dramatic reduction, but alas it is still computationally infeasible.

Assuming 200 CMs, it takes approximately 11 million operations to calculate $J_{P,M}$ for a 16 profile set on a 4K FFT channel. Optimistically, a current top-end Core i7 processor could evaluate 20,000 profile sets per second, or approximately 10^{10} profile sets per day.

Even with significant simplifications, the brute force approach is not realistic.

Grouping CMs

An optimal single profile for a group of CMs can be determined very simply by finding the minimum bit-loading capability for each subcarrier across the group of CMs. In other words, finding the column-wise minimum of the subgroup's bit-loading capability matrix, M_1, where the rows of M_1 represent the bit-loading capability vectors of the CMs in the subgroup.

Thus the problem of finding the optimal set of profiles for a population of CMs can be considered a problem of finding the optimal groupings of CMs.

Profile Coalescation Algorithm

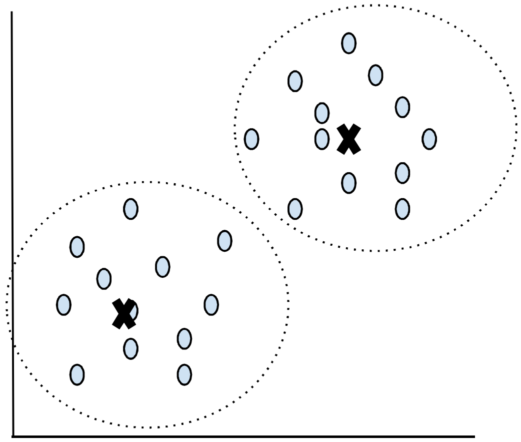
We have experimented with a technique for finding good groupings of CMs that operates as follows. Begin with one profile per CM, where each CM's profile exactly matches its bit-loading capability vector. This set of N profiles is optimal for the population of CMs; however, unless N is less than 16, this is more profiles than the CMTS can support. Next, calculate $J_{P,M}$ for this set of N profiles.

In order to reduce the number of profiles, generate the optimal set of N-1 profiles by combining two of the profiles into a single profile. The optimal value can be found by evaluating the effect of combining each of the N-choose-2 possible pairs of profiles (19,900 pairs for a population of 200 CMs) and then of these potential choices, coalescing the pair that provides the highest value of $J_{P,M}$.

This process can then be repeated to find the best set of N-2 profiles, then N-3, N-4, etc. until achieving the desired number of profiles. With large values of N, this results in a significant amount of processing ($O(N^2)$). This processing load can be reduced by pre-identifying clusters of CMs, and then starting with one profile per cluster instead of one profile per CM.

Clustering and K-Means

Clustering algorithms take an unlabeled data set and identify "clusters" of elements that are close to one another by some distance metric. The K-means [1] algorithm is a popular and widely used clustering algorithm.



K-means Algorithm for K=2 , separates the data points into two clusters, each point being assigned to the cluster whose cluster centroid (x) to which it is closest

Figure 1 - K-means Algorithm

The K-means algorithm takes a set of points and attempts to identify the set of K clusters for which the average distance of each point to the centroid (mean) of its cluster is minimized. The distance metric used in K-means could be simple L1 or L2 norm, or the Mahalanobis distance [2]. Implementations of K-means typically take as input the number of clusters (K) to split the input data set into, and return the set of K centroids as well as a set of assignments of points to clusters. K-means is not guaranteed to find the optimal set of K clusters, and in fact it will always converge to a local optimum. But, it is extremely efficient, and so it is possible to run it multiple times and select the best result.

<u>K-means Algorithm</u>

1. Select K, the number of clusters.
2. Initialize clusters by picking K random starting centroids
   - Typically this is done by selecting K random points from the set
3. Batch Update phase:
   - Assign every point to its nearest cluster (i.e., centroid to which it is nearest)
   - Once all points are assigned, recalculate centroids of the K clusters.
   - When the average distance no longer changes, move to the online phase
4. Online Update Phase:
   - Taking one point at a time, evaluate whether reassigning that point to another cluster (and then recalculating centroids) will reduce the average distance, and if so, reassign it to whichever cluster results in the smallest average distance (and recalculate the centroids).
   - When the assignments no longer change, the algorithm has converged

<u>K-Means Applied to Finding Optimal Set of Profiles</u>

For the profile optimization problem, we have experimented with using the K-means clustering algorithm to find groups of CMs that have similar bit-loading capability vectors, under the assumption that the profiles generated by the K-means clusters would provide an attractive set. This is motivated by the intuition that tightly clustered CMs would be good candidates to assign to a single profile.

The goal of K-means, however, is not directly aligned with our objective (optimizing $J_{P,M}$), and so the set of clusters found by K-means (even if it were to identify the optimal set) is not necessarily the optimal grouping of CMs with respect to our objective function.

In an attempt to sidestep this issue, rather than utilizing K-means to identify profiles, we use it as a pre-processing step for the profile coalescation algorithm to reduce the number of starting profiles, which significantly reduces the computational complexity.

K-MEANS COALESCATION ALGORITHM

Our algorithm, which we refer to as the K-means Coalescation Algorithm (KCA), runs in two stages. The first stage involves running several (e.g., 30) iterations of the K-means algorithm with K=20, and selecting the best result - that is the cluster assignment that results in the best set of profiles, as determined by its $J_{P,M}$ score. The second stage then uses this set of profiles as the initial set for running the profile coalescation algorithm.

RESULTS

CM Data

We created a synthetic data set of N=200 CMs , for a OFDM channel with N_sc=3800 subcarriers, by generating a N-by-N_sc matrix M describing the bit-loading capabilities of each CM. The CMs were divided into five clusters, with the CMs in each cluster having their bit-loading capability values chosen from a normal distribution with standard deviation of 1 and a mean unique to the cluster. The values were quantized (rounded) to integers and limited to the range (4 - 14) of bit-loading values. The mean values for the five clusters were 14, 13, 12, 10 and 9. The distribution of bit-loading values within each cluster are shown in Figure 2.

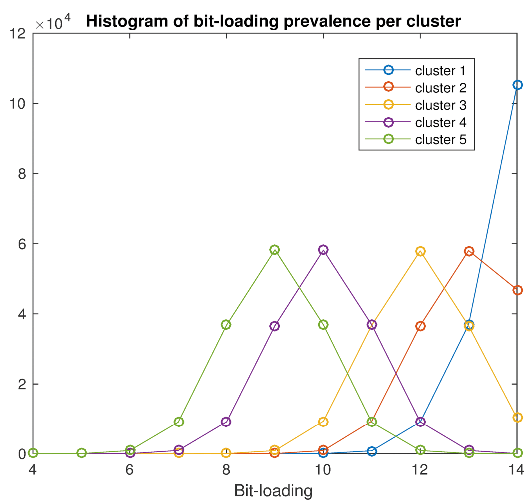


Figure 2 - Synthetic CM Data Set

Profile Coalescation Algorithm & K-means Algorithm Results

The K-means algorithm starts with a randomized set of centroids, and the results vary depending on the starting conditions. To get a sense for how this variation will affect the $J_{P,M}$ metric, we ran K-means 1000 times, and plot the histogram of the results (Figure 3).

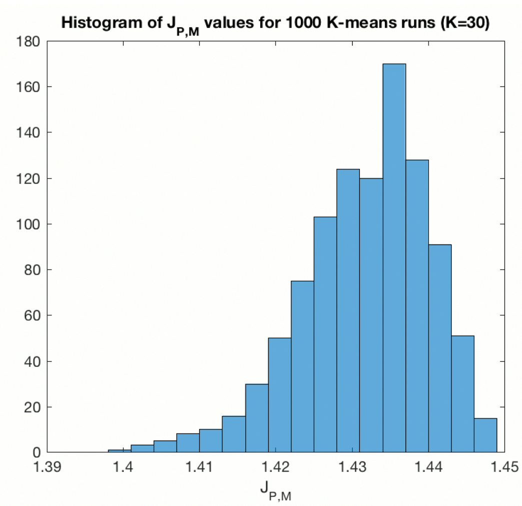


Figure 3 - Variability of the K-means Result

For the remaining profile generation algorithms, we run K-means 30 times and take the best result (as measured by $J_{P,M}$) of the 30 runs. Figure 4 shows the result of running K-means on its own to identify the profiles. The result is compared to the result obtained by the PCA algorithm. Running the full PCA algorithm on this modem population was extremely time consuming, but does provide better results than using K-means on its own.

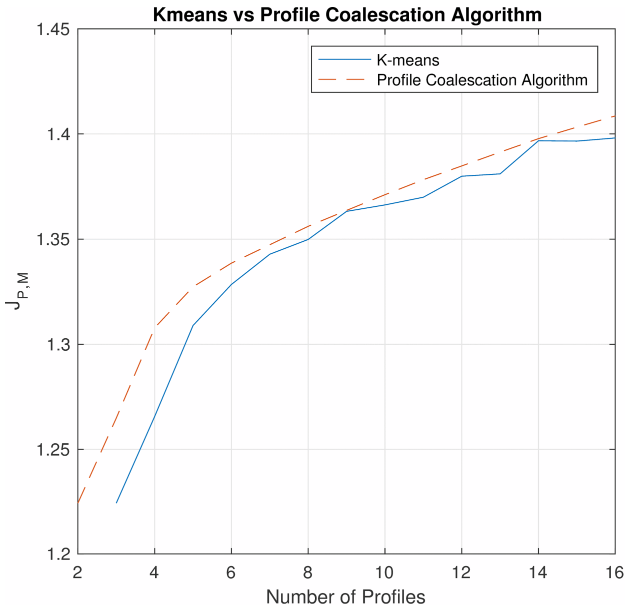


Figure 4 - K-means Compared to PCA

K-means Coalescation Algorithm Results

Our K-means Coalescation Algorithm (KCA) provided results that converge to the PCA results for our CM population. Figure 5 shows the results of the KCA algorithm using values of K from 5 to 125, compared against PCA (dashed line). In this figure, the x's show the $J_{P,M}$ value achieved by the K-means phase, and the solid line shows the progression of the coalescation algorithm.

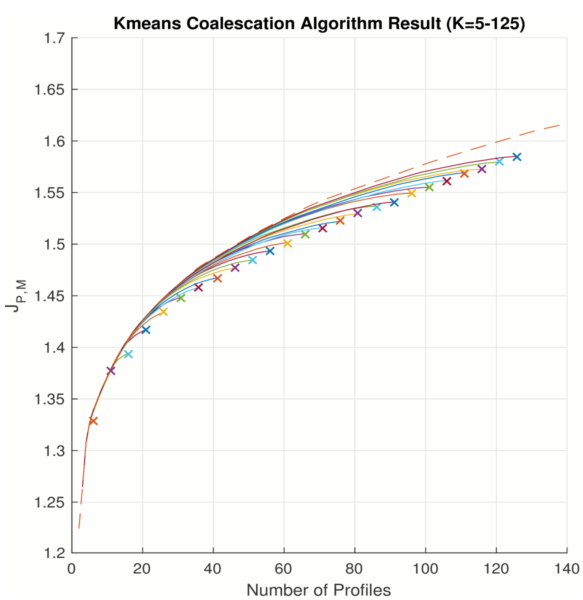


Figure 5 - KCA Results

Zooming in to see the region of most interest (where the number of profiles is ≤ 16), we can see that running KCA with a K value of 20 (i.e. 21 total starting profiles) results in $J_{P,M}$ values equivalent to the PCA for the 2-16 profile range, but with dramatically reduced computational complexity.

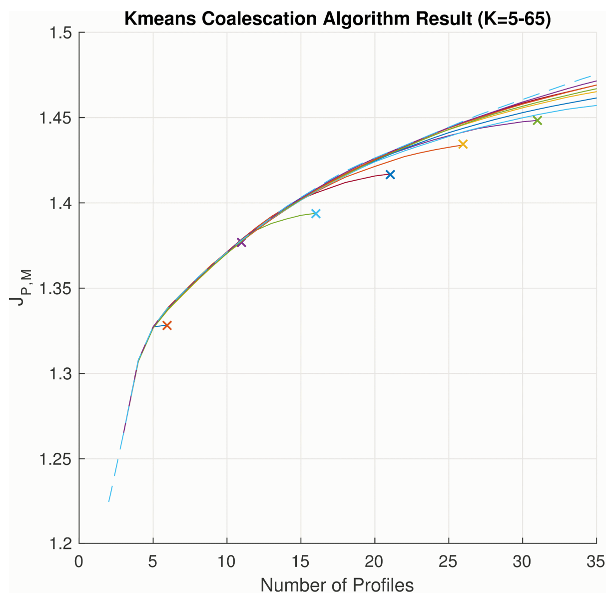


Figure 6 - Zoom of KCA Results

The gain provided by utilizing 16 profiles here is approximately 1.41, or 41% additional capacity. This result is, of course, strongly dependent on the CM population, and so could be much more or much less depending on how disparate the channel conditions are across the CM population.

FUTURE WORK

Our work at present has relied solely on synthetic data sets. In addition to the items mentioned previously, future work includes studying (and possibly refining) the KCA approach using real CM data.

Earlier, we commented that the D3.1 channel capacity is the harmonic mean of the individual CM single-user capacities. It is important to note that the harmonic mean is dominated by the minimum of the data. So, operators can expect that overall channel capacity in D3.1 is going to be strongly influenced by the worst performing CMs. Furthermore, with a profile management application in place, if the SNR for the worst performing CM degrades, the profiles can automatically adjust to maintain connectivity, eliminating any overt signal that the degradation has occurred. Unless the operator has systems in place to detect this condition, they may be disappointed by the performance of their D3.1 channels. Monitoring the bit-loading of Profile A, and identifying for remediation the CMs assigned to it will likely be an important part of D3.1 plant maintenance.

Other areas for exploration include seeking answers to the following questions. How often does a profile management application need to recalculate the profile definitions? How does a profile management application handle sudden changes in the plant conditions, which make some profiles inappropriate for a set of CMs? When is it appropriate to alert the operator to degraded CMs that are requiring an extremely inefficient profile? This analysis will help an operator develop a solid profile management application and help maintain an efficient and optimized plant.

CONCLUSIONS

In this paper we presented a formulation of the DOCSIS 3.1 downstream modulation profile optimization problem, where we showed that the overall downstream capacity can be calculated as the harmonic mean of the individual CM's single-user channel capacities (i.e. the capacity each CM would achieve if it were the only CM on the channel), and that the profile optimization problem is akin to a clustering problem.

We applied a commonly utilized clustering algorithm, known as K-means, and showed its value and performance in identifying modulation profiles. We also developed a novel approach, which we refer to as the Profile Coalescation Algorithm, which provides better performance than K-means, but comes with significantly greater computational burden. Finally, we described a hybrid of the two approaches, which we refer to as the K-means Coalescation Algorithm, which can provide equivalent results compared to the PCA algorithm, but with much lower computational complexity.

REFERENCES

[1] K-means (Lloyd algorithm): Lloyd, Stuart P. (1982), "Least squares quantization in PCM", IEEE Transactions on Information Theory 28 (2): 129–137, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.1338

[2] Mahalanobis, Prasanta Chandra (1936). "On the generalised distance in statistics" (PDF). Proceedings of the National Institute of Sciences of India 2 (1): 49–55.

[3] DOCSIS 3.1, Physical Layer Specification, CM-SP-PHYv3.1-I08-151210, December 10, 2015, Cable Television Laboratories, Inc.

[4] DOCSIS 3.1, MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I08-151210, December 10, 2015, Cable Television Laboratories, Inc.

[5] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201. Retrieved 2009-04-07.