VIRTUAL EXPERIENCE
OCTOBER 11-14

2021 Fall Technical Forum
SCTE • NCTA • CABLELABS

Optimizing DOCSIS 3.0 Configuration in the Upstream through Applied Reinforcement Learning.

A Technical Paper prepared for SCTE by

Kevin Dugan
Scientist 3, Enterprise Data Analytics & Data Intelligence
Comcast
1800 Arch Street, Philadelphia, PA 19103
719.493.2600
kevin_dugan2@cable.comcast.com

Maher Harb
Director, Data Science
Comcast
1800 Arch Street, Philadelphia, PA 19103
267.260.1846
maher_harb@comcast.com

Dan Rice
Vice President, Access Architecture and Technology
Comcast
1800 Arch Street, Philadelphia, PA 19103
720.512.3730
daniel_rice4@comcast.com

Robert Lund
Principal Engineer
Comcast
1800 Arch Street, Philadelphia, PA 19103
303.907.9690
robert_lund@comcast.com

Table of Contents

Title **Page Number**

1. Introduction 4
2. Reinforcement Learning Design for PMA Systems 4
   2.1. RL Concepts for US PMA 4
   2.2. RL Applied to US PMA System 6
      2.2.1. State 6
      2.2.2. Actions 7
      2.2.3. Reward System 7
      2.2.4. Policy Updates 8
3. System Architecture 8
   3.1. Lab System 8
   3.2. Data Pipeline 9
   3.3. Closed Loop 10
4. Building Dynamic Policies 10
   4.1. Initial Policy from Historical Data 10
   4.2. Tuning the Initial Policy 11
   4.3. Multiple Policies 11
5. Performance Study 11
   5.1. Design 11
   5.2. Policy Behavioral Expectations 12
   5.3. Performance Evaluation 12
      5.3.1. Static Policy 12
      5.3.2. Dynamic Policies 16
      5.3.3. Summary Evaluation 20
   5.4. Opportunities for Enhancement / Potential Future Steps 22
6. Conclusion 22

Abbreviations 24

Bibliography & References 25

List of Figures

Title **Page Number**

Figure 1 - RL Framework of the PMA System. 5
Figure 2 - Example of a Unique State Id 7
Figure 3 - Lab System Architecture 9
Figure 4 - Data Pipeline Sequence 9
Figure 5 - Ordering Historical Data for Learning RL Policy 11
Figure 6 - Static Policy Profile Transitions in Terms of Speed 13
Figure 7 - Bonding Group Speed Distribution 13
Figure 8 - UCCW Rates Encountered (Static Policy) 14
Figure 9 - 4-Channel Upgrade Trajectory from Baseline (Static Policy) 14
Figure 10 - 6-Channel Upgrade Trajectory from Baseline (Static Policy) 15
Figure 11 - Profile Type Utilization by Frequency (MHz) on Static Policy 15
Figure 12 – RPD Profile Speed over Time (Dynamic Policies) 16
Figure 13 - Profile Speed Distributions by Dynamic Policy 17

Figure 14 - UCCW Rates of RPDs by Dynamic Policy 17
Figure 15 - Changes in Profiles with UCCW Rates on RPD HS3 18
Figure 16 - 4-Channel Upgrade Trajectory from Baseline (Dynamic Policies) 18
Figure 17 - 6-Channel Upgrade Trajectories from Baseline (Dynamic Policies) 19
Figure 18 - Profile Type Utilization by Frequency (MHz) for Dynamic Policies 20
Figure 19 - Profile Speed over Time, Static vs Dynamic 20
Figure 20 - Dynamic Policy Learning Better Action 21

List of Tables

Title **Page Number**

Table 1 - Attributes of a State 7
Table 2 - Lab Devices 8
Table 3 - Static Policy Profile Speed Metrics 15
Table 4 - Dynamic Policy Information 16
Table 5 - Dynamic Policy Raw Profile Speed Totals 19
Table 6 - Profile Speed Metrics, Static vs Dynamic 21

List of Equations

Title **Page Number**

Equation 1 – SARSA Trajectory 5
Equation 2 - TD SARSA Equation for Value Updates 5

1. Introduction

In 2020, Comcast deployed a Profile Management Application (PMA) system (1) for optimizing the DOCSIS 3.0 (D3.0) configuration across upstream channels in the network to achieve the proper balance between efficiency and robustness (fault tolerance). The established PMA system strengthens organizational objectives to increase network capacity, proactively respond to impairments, and support robust service optimizations for customers. At its core, the current system adopts a rules-based approach, in which a static policy (in the form of defined thresholds) for the different telemetry features (e.g., signal-to-noise ratio and codeword error rates) govern the choice of channel configuration.

Limitations within PMA exist when shaping telemetry thresholds to adapt to a wider range of environmental conditions. Currently, configurations are assigned to channels starting with conservative profiles and progressively moving toward efficient, yet less robust profiles. Further innovations within Comcast's PMA implementation focus on the delicate balance of applying intelligent, dynamic decision-making policies while preserving proper configurations for the diverse set of network devices.

A reinforcement learning (RL) approach for PMA allows, through experience, learning an optimal policy and therefore, enhancing the criteria used at various decision points. Simultaneously, RL simplifies policy management by consolidating permutations of telemetry boundaries into a single entity, called a 'state'. PMA efficacy improves with RL by reducing the latency of transitioning into optimal, efficient profiles, and doing so with increased confidence across varying network conditions (4). Inherent in this implementation is the risk reduction for operators to deploy more profile changes that maximize capacity without crossing the boundary that introduces disruption in service. This paper introduces a proof-of-concept RL-based PMA system along with performance study based on initial experimentation conducted in our laboratory.

2. Reinforcement Learning Design for PMA Systems

The purpose for implementing RL on upstream PMA is twofold: to manage PMA using a dynamic policy that continuously learns and to select optimal profile configurations with increased efficiency. An RL-based system provides the framework to facilitate these two objectives. A dynamic policy that is updated over each time step is more attuned to fluid network conditions than a static policy with fixed global thresholds. When a dynamic policy is updated, the decision criteria for choosing the optimal profile configuration under the current network conditions represents the best-known policy discovered by the system (2).

2.1. RL Concepts for US PMA

The RL sequence consists of an *agent*, selecting *actions* to take from a given *state*, and calculating the *value* of the action taken with respect to a *reward* system as the resulting state interacts with the *environment*. Figure 1 represents the sequence of interactions in a feedback loop. As the agent collects the rewards, it updates the policy using the *value function* (Equation 2) that satisfies the Bellman Equation (2) to continually refine the values for states-action pairs encountered. This process is called *value iteration* and is the basis of the dynamic policy.

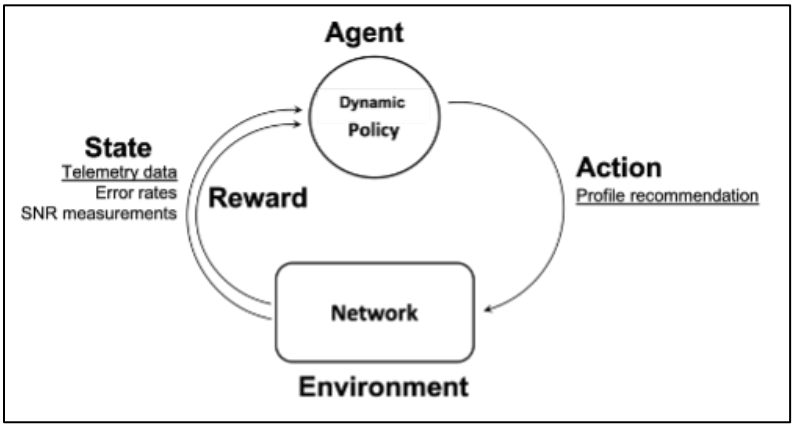

Figure 1 - RL Framework of the PMA System.

State-action values are numeric representations describing how valuable it is to take a given action from any state. They are calculated with respect to the next state-action pair, establishing the update process for sequential decision-making as Markov Decision Process (MDP). As the system builds experience, convergence occurs in the policy whereby updates to state-action pairs change the values over time in progressively smaller increments. The calculations of state-action values are based on the acronym SARSA. S, A, and R respectively denote State, Action, and Reward. With a subscript referring to the time step, SARSA can be represented with the following trajectory:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots \quad (1)$$

In an MDP, the consequences of an action taken within the current state do not depend on maintaining the entire history of the trajectory; instead, just updating the state-action pair encountered maintains the legacy of experience already gained. The nature of the update considers how rewards are weighed, which influences how the policy is learned. For the proof of concept described in this paper, the Temporal Difference (TD) SARSA equation, $Q_\pi(s, a)$ in Equation 2 for online policy improvement is used to update the policy. The hyperparameters, α and γ, influence the learning rate and value of the future reward, respectively.

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{(t+1)}) - Q(S_t, A_t)] \quad (2)$$

The α and γ terms are both floating point values between zero and one. α is a fixed learning rate where a value close to zero slows the learning and closer to one increases the learning rate. γ is a discount rate that determines how much to weigh immediate rewards and potential future rewards. As γ approaches one, the value of potential future returns influences the state-action pair estimates just as much as the immediate return; whereas a value closer to zero treats near-term rewards with more emphasis than future rewards. With respect to the US PMA effort, responding quickly to poor telemetry is critical in reducing customer impact. Finding an appropriate combination of these hyperparameters is an important objective for the problem at hand.

A dynamic policy is interesting to the PMA problem for its adaptive behavior that adjusts state-action pair values, and then uses those new values in real-time. With a dynamic policy in place, the RL system can adjust decision criteria under clean or adverse conditions. Consider a scenario where network impairments are causing high Uncorrectable Codeword error ratio (UCCW) rates and low signal-to-noise ratio (SNR). The RL policy, already having been exposed to this state, will recommend a profile change toward alleviating the side-effects of the larger problem (UCCW rate). If the issues persist, it is possible

for the RL policy to learn to take more aggressive action. The negative reward associated with the more conservative step will reduce its state-action value. Once an acceptable profile is reached, the state-action value is positively updated. At a point, if the cycle continues, the value of taking the aggressive action will become higher than the conservative action. Thus, the agent will then choose the aggressive action next time the poor telemetry is encountered.

In terms of α and γ, having parameters that reflect values associated with learning quickly is an intuitively logical approach. When an impairment occurs, the quicker the policy can adapt within the environment, the better decisions it will make. An example would be to set α to 0.8 and γ to 0.3. In this paper, multiple variations of the policy parameters are explored to inform of system short-term and long-term behavior implications associated with different policies.

2.2. RL Applied to US PMA System

Within the scope of an upstream PMA system, the states, actions, and a reward system were defined to represent characteristics of the current telemetry and configuration data to choose the optimal profile configuration for the present conditions (4). A state represents a distinct set of channel configurations and their associated network metrics. Each defined action is available for each state. The reward system is the mechanism used to influence how valuable it is to take an action from a state. As the system iterates over the timesteps, rewards are used to update the policy for the current action.

One key principle in RL systems is the trade-off between exploration and exploitation. Exploitation occurs when the highest-valued action previously encountered is taken from a state. However, that may not be the absolute best action to take for that state. To find the optimal action, the algorithm could select a random action to evaluate. When operating in production systems, random exploration is a risky endeavor because the policy could choose a transition that either introduces elevated UCCW rates, or transition to a slower profile when it is completely unnecessary to do so. In this POC, the evaluation of dynamic policies omits the exploration step and uses only exploitation to select the best actions available.

2.2.1. State

In the RL system for this paper, a state is considered a collection of telemetry metrics and configurations that are represented by discrete bins, with each state being unique. Continuous variables, like telemetry, were categorized by value ranges. Categorical variables fell naturally into their associated bins. Table 1 provides a breakdown of attributes to bins. In reality, only a small portion of the possible states will be encountered. As an example, not all CMTSs may be configured with six upstream channels and will never encounter any state where a fifth or sixth channel is represented.

A unique state is represented as a concatenated string of the bins as shown in Figure 2. In this example, the state of the channel on Remote PHY Device (RPD) HN1 is interpreted as a sub-optimal profile configuration that is experiencing a poor UCCW rate and needs to be downgraded to a more robust profile configuration.

Table 1 - Attributes of a State

Category	Attribute	# Bins
Telemetry	Uncorrectable Codewords (UCCW)	3
	Correctable Codewords (CCW)	3
	Signal to Noise Ratio (SNR)	6
Channel Configuration	Channel Width	3
	Modulation	5
	Profile Type	5
	Traffic Volume	2
	Channel Frequency	6
	CMTS	5
	Total # Possible States	243,000

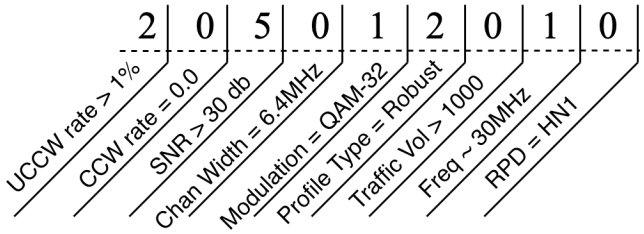

Figure 2 - Example of a Unique State ID

2.2.2. Actions

For each defined state, the complete set of actions is available to choose from when making profile recommendations. Like the states, many actions will never be encountered for certain states. For example, if the upstream channel was running on the optimal profile, there is no ability to move to a more efficient profile. Therefore, the state-action values for upgrades would all remain at zero.

In this RL system, three categories of actions were defined:

1. Upgrade or downgrade
2. Remain in the same configuration
3. Transition onto and off the most robust configurations, referred to as 'transient' profiles (these are designed to deal to dynamic impulse or burst noise)

Actions were limited to a maximum of four steps to restrict the dynamic policies from taking large action steps which would increase the likelihood of entering a poor state.

2.2.3. Reward System

The reward function serves to describe to the agent how it ought to behave. The key attribute to direct the agent in the simplest way is the UCCW rate. In this implementation, the reward system is boiled down to an evaluation of a Boolean condition that answers the question, "Is the UCCW rate greater than 1%?". If it is, a large negative reward (punishment) of -10 is given to the agent. Otherwise, the reward is $1 + profile\ speed\ gain$. A negative reward can still be observed if the UCCW rate is below 1%. For example, if a channel is in an optimal profile and the algorithm moves it to a transient profile, the loss of

speed is greater than the static bonus of +1 for keeping the UCCW rate low. This simple approach is designed to encourage upgrades in good conditions, and downgrades in poor conditions.

2.2.4. Policy Updates

The heart of the dynamic policy is the process of value iteration. Value iteration using TD SARSA from Equation 2 is the update algorithm that continues to adjust the state-action values until convergence (2). For each iteration, and each state and action encountered, the Bellman equation (2) is applied as an update rule in the form of Equation 2 and the profile recommendation logic uses the freshly updated policy in the same iteration.

3. System Architecture

3.1. Lab System

The lab system architecture was built off the existing Profile Management Application (PMA) Lab design. The primary functional groups consist of:

1. Impairment generation
2. RF switching matrix
3. Diplexing/combining components
4. Cable modem racks
5. Traffic generation
6. RPD nodes under test
7. vCORE and backoffice

The automation controller randomizes impairment profiles on the generator via an SCPI interface. It is also used to run GNURadio to create bespoke waveforms played back through a software-defined radio. The impairment sources are connected to an 8 x 16 port RF switch to steer or distribute the impairments to the appropriate devices under test. The impairments are combined into the appropriate points to feed either the RPD upstream burst receiver or the cable modems' downstream receivers. For the traffic generation loop, the network side interface is connected to the Distributed Access Architecture Switch (DAAS) and the CPE ports are VLAN'ed and connected to each cable modem on a high-density, mobile rack.

Five RPDs, with an average of 8 cable modems (CM), made up the population of devices for the trial. Each RPD contains a bonding group of either four or six D3.0 upstream channels, with a total of 24 upstream channels across the RPDs. Random impairments (none to severe) were introduced to evaluate the policies under both clean and adverse telemetry.

Table 2 - Lab Devices

RPD Name	Number of Channels	Number of CMs	Vendor
AS2	6	6	CMTS X
AN1	4	8	CMTS X
CN1	4	7	CMTS Y
HN1	4	12	CMTS Z
HS3	6	8	CMTS Z

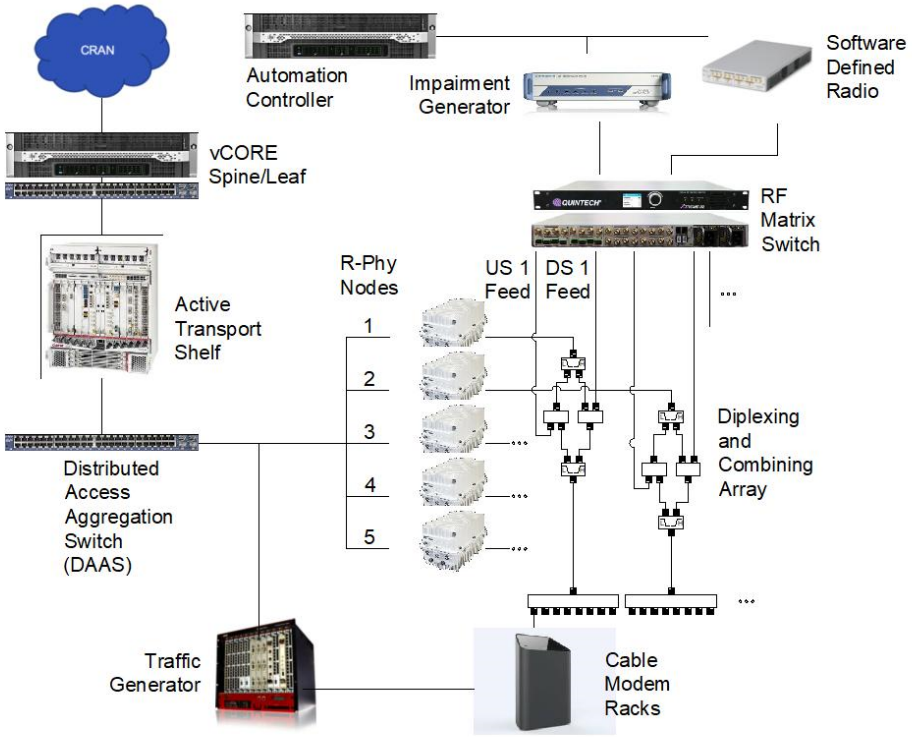

Figure 3 - Lab System Architecture

3.2. Data Pipeline

Each section of the pipeline is decoupled from the other sections to accommodate for experimentation and manipulation of policy parameters, states, actions, or other logic that would benefit from incremental changes. The sequence of the pipeline is shown in Figure 4:

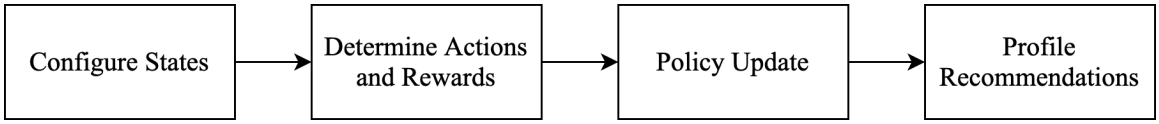

Figure 4 - Data Pipeline Sequence

The primary inputs are the attributes that make up a unique state (described in Section 2). Once collected, the current state can be assigned for each channel on the RPD. The data is associated with a timestamp that can be used, if necessary, to retrain a policy through the sequence of actions taken using historical data.

Arranging the data points by timestep for each upstream channel, the sequence of actions taken, and states encountered in order can be observed and rewards assigned. Being an independent step in the pipeline, this gives the opportunity to update the reward function or actions, and to be able to run the updates over the complete set of historical data or just the latest data points that need updated. In this pipeline, acquiring the action taken is not solely dependent upon the profile recommendation from the previous step, which makes determining the action taken delayed. A separate transaction manager must apply the recommendations first, and that function may or may not have succeeded.

In the policy update step, the latest timestep that needs to run through the SARSA equation with the full suite of the current state, action, reward, next state, and next action data is collected. The collection of all updated state-action pair values is preserved for use in recommending profiles.

Making profile recommendations is relatively trivial after the policy has just been updated; however, there are special cases to account for to either prevent abnormal behavior or change the degree of exploration desired. In most cases, the agent simply looks for the highest-valued action for the current state, also known as exploitation. Exploration is the selection of a random action a specified percentage of the time. In a production environment, either the exploration is very limited or not used at all. The profile recommendations in this paper for RL-based policies were all derived using exploitation – selecting the highest-valued action in all cases.

It is possible for the RL system to encounter states not seen before. When this happens, conventional logic is to select a random action. This could be detrimental to the channel's capacity on live CMTSs. In this described RL system, logic was added to directionally influence the profile recommendation based on the UCCW rate for the channel if a state is encountered for the first time. There is no need to downgrade a channel's profile if the telemetry is obviously adequate for consideration of upgrading.

3.3. Closed Loop

The closed loop system consists of the lab architecture and operation, the data pipeline, and the profile transaction manager. The lab generates telemetry and maintains configurations, which is processed through the pipeline, and the profile recommendations are applied onto the systems. Pipeline cycles are scheduled to complement the noise transition schedule in the lab by executing just prior to the noise transitions, ensuring that the next profile recommendations are based on the most current telemetry.

4. Building Dynamic Policies

Training a dynamic RL policy from an absolute baseline in this problem space would require more iterations over the pipeline than is realistically feasible to allow for adequate exploration of the state-action space in a live feedback loop. An approach to priming the state-action pair values in the policy is to leverage historical actions from the existing static policy. Starting from this point enables the policy tuning to occur in less timesteps, while also allowing for liberal exploration of the state-action space.

4.1. Initial Policy from Historical Data

The approach in using historical data for initially creating a RL policy follows concepts from imitation learning (IL) and inverse reinforcement learning (IRL). Both disciplines use a demonstration - a replication of behavior sequences in the problem space – to acquire experience and learn a policy. "The inverse reinforcement learning problem is to find a reward function that can explain observed behavior" (3). With a reward system in place, determining the right policy is a matter of exploring values of α and γ for the TD SARSA equation.

The static policy represents the human-generated data, taking directionally accurate (not necessarily optimal) actions under various network conditions. Doing so approximates state-action pair values toward their true values, which are refined over future value improvement iterations. The historical data is transformed into an ordered set of actions associated to states, as in Figure 5.

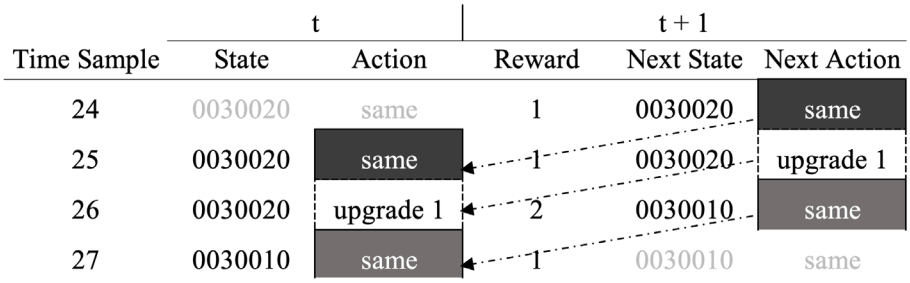

Figure 5 - Ordering Historical Data for Learning RL Policy

For each row, the $S_0, A_0, R_{t+1}, S_{t+1}, A_{t+1}$ observations needed to update the policy are arranged by windowing over the data and observing the sequence of profile transitions. This arrangement of historical data is the model used in future cycles when training RL policies using any decision-making policy. With the de-coupled implementation of the data pipeline, each step of data collections and transformations can be re-processed in their entirety. This becomes useful when adjusting a reward function or making any changes to the state or actions.

4.2. Tuning the Initial Policy

To build a true RL policy from the static policy's historical data, the RL policy itself is used to make profile recommendations and receive the resulting state condition through tens of thousands of state encounters. Over the iterations, the policy learns the highest-valued actions to take for many common situations using progressively restrictive variations in the trade-off between exploration and exploitation.

Initially, the policy was allowed to explore 100% of the time with the goal of starting to refine the state-action values. The exploration variable was decreased incrementally over time to the point where no exploration occurred to allow for the policy evaluation.

4.3. Multiple Policies

For each iteration, variations of policies can be trained on the actions taken by another policy. Values of α and γ are permuted to build multiple policies simultaneously. Since the parameters influence learning rate and reward weighting, the permutations of the parameters build policies that behave differently from one another, but still directionally appropriate.

5. Performance Study

The performance study focuses on comparing the static policy and the dynamic policies; specifically, evaluating profile speeds, profile sequences used, and policy responses to impairments. The objective is to identify a dynamic policy that can match or exceed the performance of the static policy.

5.1. Design

Every policy in the study will manage the profiles for five RPDs having either four or six upstream channels between 10.4MHz – 40.5 MHz along the spectrum. A minimum of 25 iterations for all five RPDs through the data pipeline will provide opportunities for the system to experience impairments from the randomized noise transitions from the lab. The dynamic policies are configured to not conduct any random exploration; they will behave in a similar manner as the static policy by always taking the best-

known action from a given state – even if the best-known action dynamically changes during policy updates.

Prior to each iteration, the profiles on all upstream channels are set to a baseline that mimics the configuration on a CMTS / RPD when it is onboarded into the PMA system. The channels lower on the spectrum begin in a transient profile configuration, while the upper 6.4MHz channels (above 27MHz) begin in sub-optimal profiles configured at modulation QAM-64. If the system has a sixth channel, it starts in a transient profile as well. For each dynamic policy pipeline run, updating the policy includes updating the other policies in consideration by applying the actions taken to different values for α and γ. In a sense, the decision-making policy is demonstrating actions to take for the other policies that get updated by evaluating decision-making policy's actions.

The four-channel configurations are all 6.4MHz wide, while the six-channel configuration adds a 3.2MHz channel below and a 1.6MHz channel above the four-channel configuration.

5.2. Policy Behavioral Expectations

Under good network conditions, the policies are expected to upgrade profiles for more efficiency and capacity until the most efficient profile is reached. The channel would ideally remain in the optimal profile if conditions are supportive. As UCCW rates climb over the 1% threshold, the policies are expected to downgrade the profiles to those more suited to handle noisy conditions. By doing so, the UCCW rate may recover at a point in the downgrade process, whereby the policy is expected to attempt periodic upgrades to check if the issue has been cleared. Otherwise, the policies are expected to learn to remain at a lower profile over time. This study was not designed to observe policy changes for long-running impairments, mostly due to time and resource constraints.

5.3. Performance Evaluation

Profile speed is a heuristic that can be used to represent the overall health of a bonding group (the collection of each upstream channels on each RPD). As the bonding group approaches the maximum profile speed, it is representative of the policy taking appropriate actions under good telemetry conditions. The maximum possible speeds are calculated by summing the optimal profile speeds for each upstream channel per system. Observing instances where maximum profile speed is not reached, determination of the cause falls into three categories: network impairments, poor decision by the policy, or an error external of the data pipeline (RPD channel error, transaction manager failure, etc).

UCCW rates are indicators where policies are expected to upgrade or downgrade profiles, if possible. Policies are expected to begin upgrading the profiles as conditions improve, downgrading under poor conditions, and remain in the best profile possible.

Achieving the best possible profile, as quickly as possible, is measurable by the number of timesteps it takes to achieve the optimal profile in a clean environment. To ensure a fair side-by-side comparison, the dynamic policies were limited to the largest transition step allowed in the static policy. The trajectory while transitioning off the baseline profiles is indicative of how assertive the policy is with respect to reaching the best-available profile.

5.3.1. Static Policy

The static policy has deterministic behavior, albeit through a combination of several thresholds. It was designed to reach optimal profiles by making transitions that are proportional to multiple telemetry metric

thresholds. Conservative decision criteria were purposefully built into the logic to greatly reduce introducing negative impact for customers.

5.3.1.1. Profile Speed Analysis

Observing the transaction history for each of the 25 timesteps, the static policy's ability to reach and maintain a steady state for the majority of RPDs indicates only few impairments were encountered. This allowed the policy to continue to use optimal profiles on many of the individual channels.

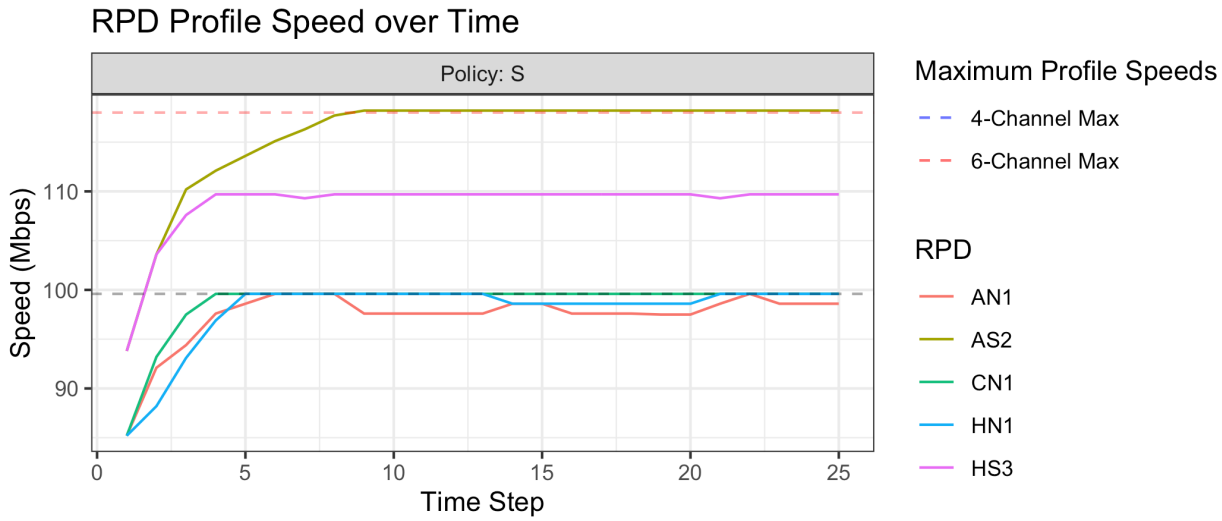

Figure 6 - Static Policy Profile Transitions in Terms of Speed

Under the static policy, the distribution of profile speed values is correlated to the UCCW rate values above and below 1%. Figure 7 affirms the behavior of the static policy achieving a steady state with mostly optimal profiles under the network conditions experienced in the iterations. AS2, CN1, and HN1 achieved the maximum possible speed for the bonding group and maintained during a significant portion of the study. A channel error occurred on HS3 that prevented the policy from achieving full speed for the bonding group. For that purpose, the policy achieved the maximum speed possible for the remaining five channels.

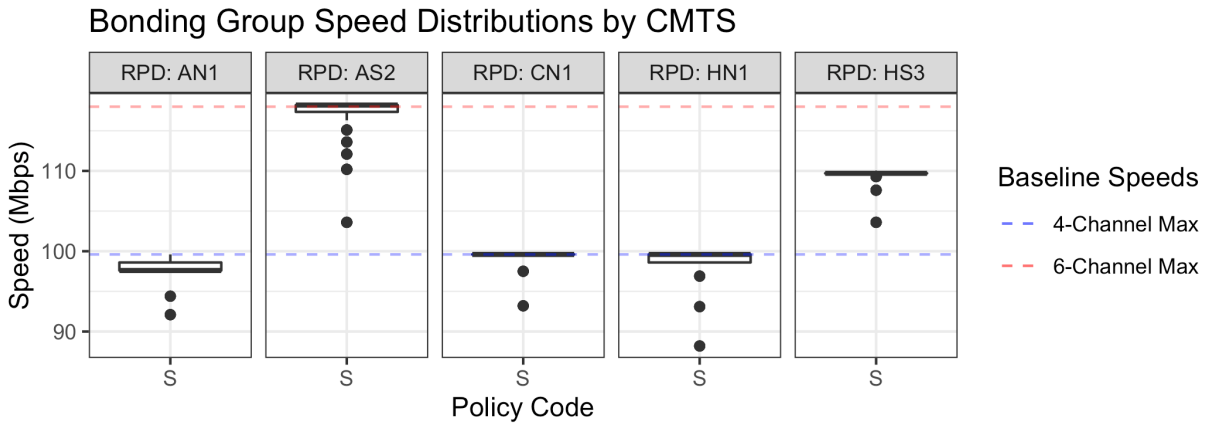

Figure 7 - Bonding Group Speed Distribution

Figure 8 illustrates how the RPDs were unaffected by the random noise generation from the lab. Despite experiencing majority clean UCCW rates, not all RPDs reached optimal profiles for all channels. A single channel on AN1 fluctuated between the top two profiles relating to reported SNR values and thresholds associated with SNR in the policy. Otherwise, the static policy behaved as expected, and under the conditions of the test environment, achieved the optimal profile available for the channel given the telemetry feedback.

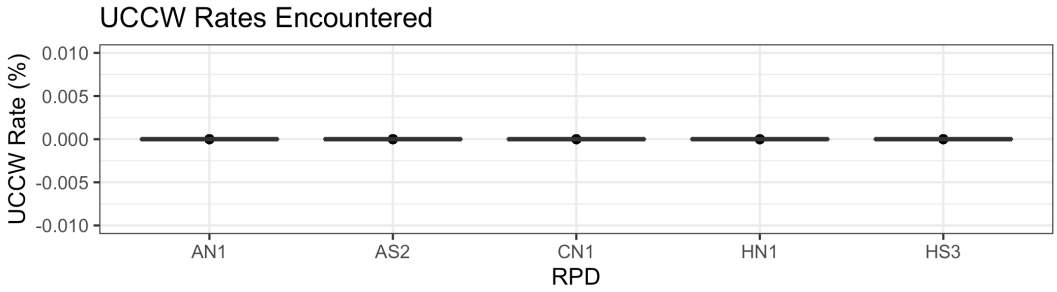

Figure 8 - UCCW Rates Encountered (Static Policy)

5.3.1.2. Latency to Optimal Profiles from Baseline

The number of timesteps the static policy took to reach a steady state operating on mostly optimal profiles for four-channel RPDs is represented in Figure 9. Omitting step zero, it took an average of four timesteps to get all three systems to a steady state on optimal profiles. Each RPD was upgraded at a slightly different rate, indicating a factor besides UCCW rate was affecting how large of a transition to make for each channel (SNR or CCW rates). The average SNR for RPDs CN1 and HN1 was below 30 dB for four of the six timesteps, while AN1 enjoyed an average SNR above 40 dB for the same timesteps. In spite of this, AN1 was the last to reach a steady state on optimal profiles.

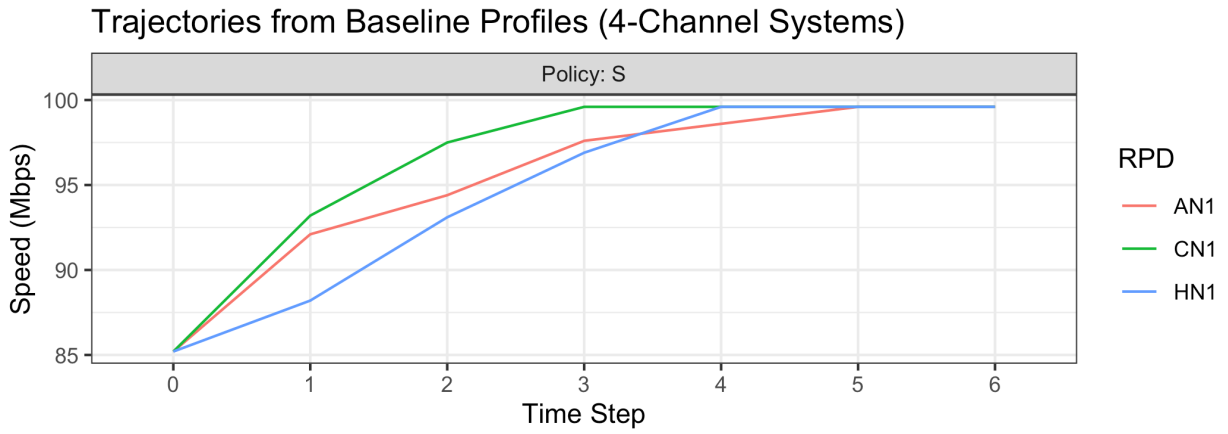

Figure 9 - 4-Channel Upgrade Trajectory from Baseline (Static Policy)

The six-channel RPDs demonstrated a wider disparity in terms of steps to reach a steady state. HS3 took two steps before leveling off, albeit notably by not upgrading the single channel that incurred reporting errors during the trial. AS2 reached a plateau after eight timesteps. This is attributed to the number of upgrades that had to occur for the upstream channel at 10.4MHz. Since it starts at the lowest possible profile, and limits upgrades to a maximum of four steps, this channel takes the longest to reach an optimal profile.

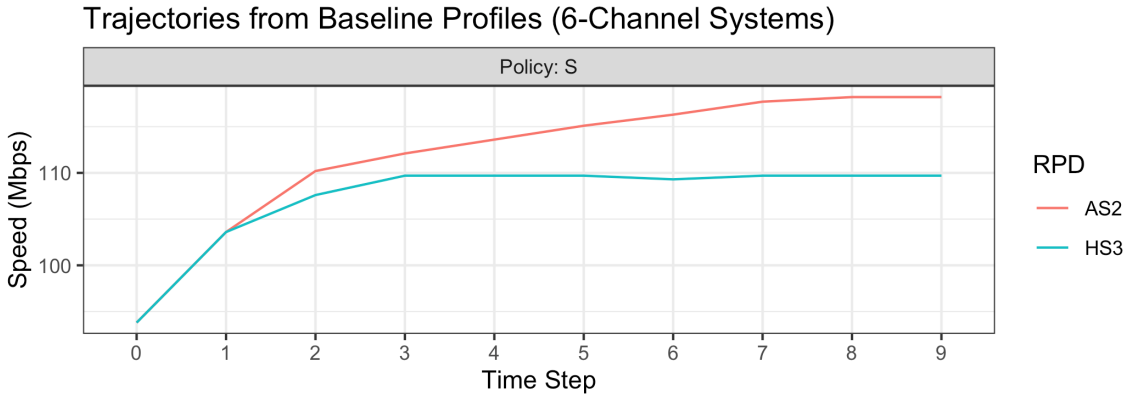

Figure 10 - 6-Channel Upgrade Trajectory from Baseline (Static Policy)

5.3.1.3. Summary Statistics

Over 25 iterations, the static policy achieved an average speed of 522 Mbps per iteration (all five RPDs), for an average bonding group (RPD) speed of 104.3 Mbps per iteration. Table 3 has the breakdown of profile usage:

Table 3 - Static Policy Profile Speed Metrics

Profile Type	% of Total Speed	% Profile Occurrences
Optimal	88.95%	85.15%
Sub-optimal	8.83%	8.44%
Transient	1.37%	2.24%
Below QAM-64	0.84%	4.17%

Also of interest is the relationship between which profiles were used in which locations along the spectrum during the trial. The transient profiles exist largely on the low end of the spectrum, as expected with clean telemetry throughout. Optimal profiles occurred often on the 6.4MHz channels from approximately 16MHz – 36MHz.

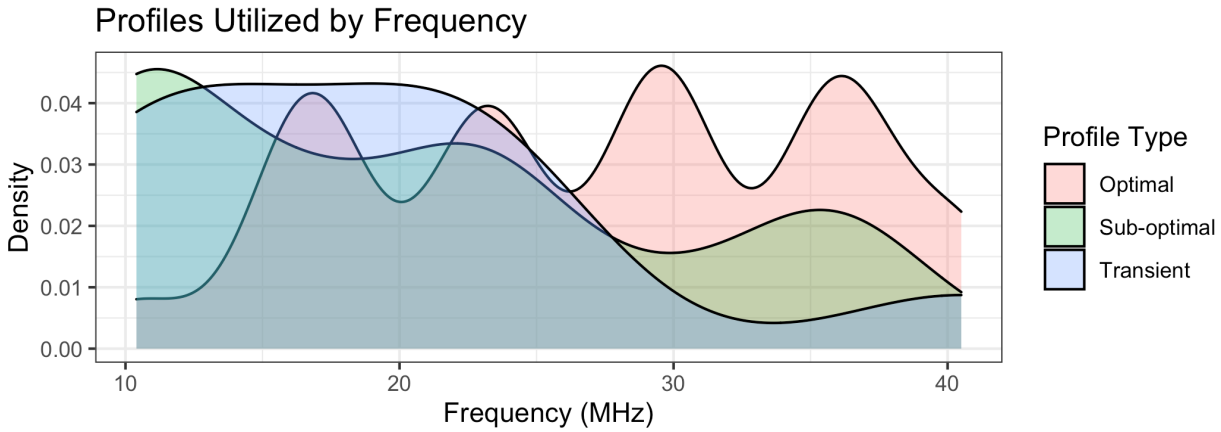

Figure 11 - Profile Type Utilization by Frequency (MHz) on Static Policy

5.3.2. Dynamic Policies

The four dynamic policies chosen for illustration in this paper were selected considering the values of α and γ that affect the learning rate and the weight placed on future rewards, respectively. Table 4 describes which values were used per policy, along with a high-level summary for how the policy generally behaved.

Table 4 - Dynamic Policy Information

Policy	α	γ	Policy Behavior
A	0.9	0.2	Achieved optimal profiles on 4 out of 5 RPDs
B	0.8	0.8	Indecisive, did not reach steady state, fluctuated profiles
C	0.3	0.8	Not as assertive as policy A, fluctuated profiles
D	0.8	0.2	Similar to policy A, optimal profiles on 3 out of 5 RPDs

5.3.2.1. Profile Speed Analysis

The dynamic policies experienced elevated UCCW rates on RPD HS3 during the trial. This prevented any of the policies from achieving optimal profiles in a steady form on that system. Policy A exhibited the most similar behavior as the static policy, maximizing the four-channel systems and reaching optimal speed on one of the six-channel bonding groups. Policies B and C have the most severe 'indecisiveness', fluctuating between profiles and largely not showing the ability to maintain a steady speed.

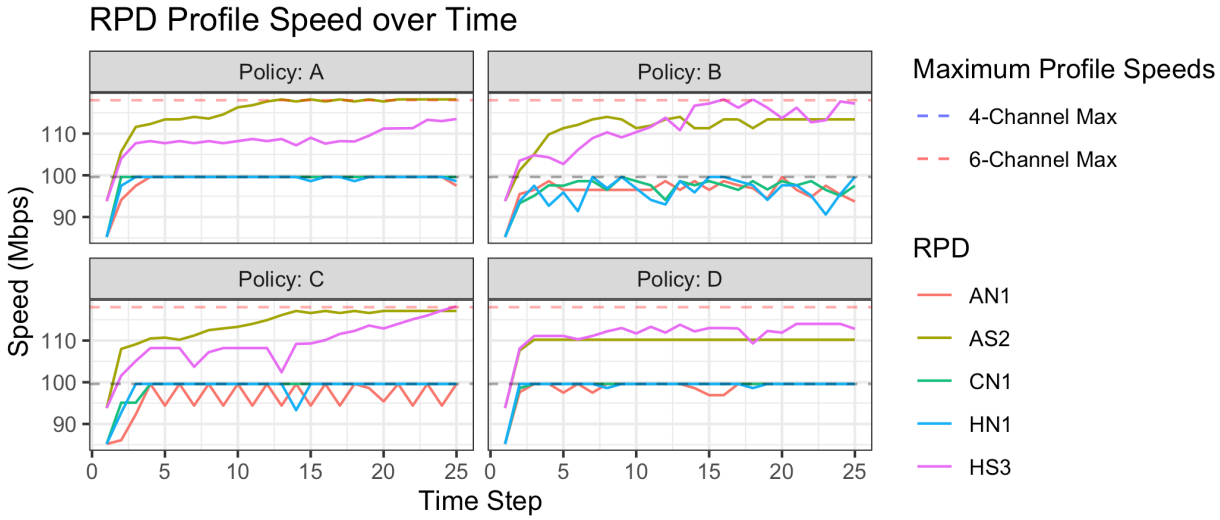

Figure 12 – RPD Profile Speed over Time (Dynamic Policies)

Figure 13 illustrates how the dynamic policies performed over the 25 iterations with respect to profile speeds distributions for each CMTS. The tight distributions observed on systems AN1, CN1, and HN1 indicate a lack of impairments over the iterations and affirm the observations in Figure 12. Certain policies quickly achieved optimal profiles and maintained throughout. The larger distributions are found in the six-channel systems, AS2 and HS3. Almost all the dynamic policies struggled to achieve consistent optimality on the two RPDs. Policy A had the most success and tended to be the most assertive of the policies.

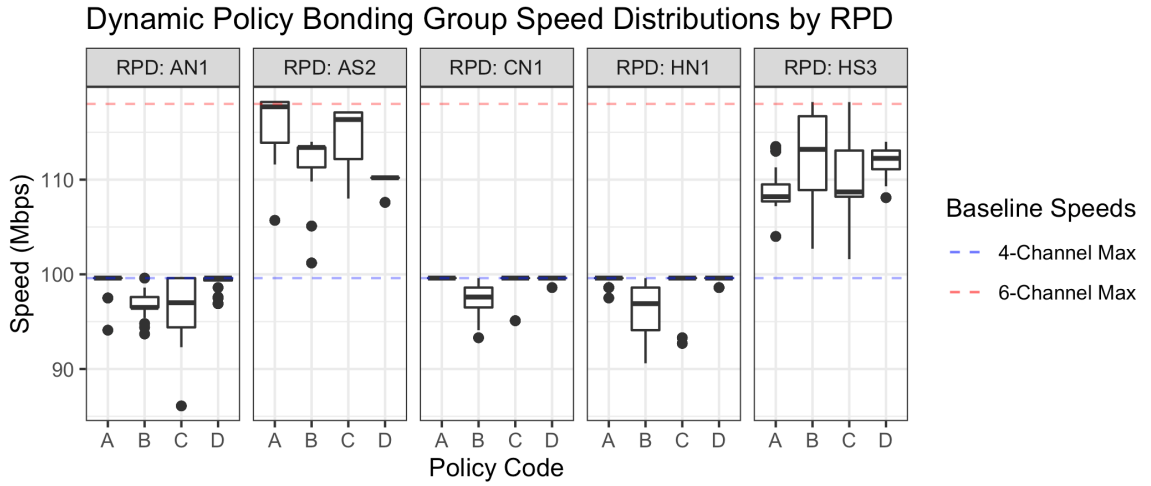

Figure 13 - Profile Speed Distributions by Dynamic Policy

UCCW rates did have an impact on system HS3 over the course of the trial for each of the dynamic policies. At one point, policy D experienced a channel with a severe UCCW rate of 100% (Figure 14). Since channels with severe impairments will not achieve the best profile, RL policies adapted to reaching profiles suitable for the impairment event.

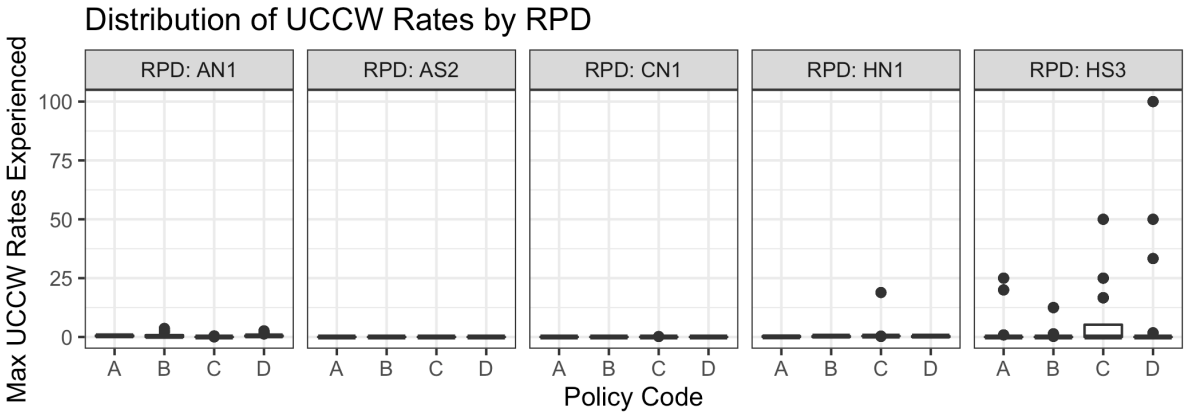

Figure 14 - UCCW Rates of RPDs by Dynamic Policy

Figure 15 highlights the UCCW rates observed on each iteration for each policy in relation to the overall profile speeds for each timestep on system HS3. In almost all cases, as a UCCW rate above 1% was detected, a decrease in profile speed for the bonding group indicates one or more channels downgraded to a slower, more robust profile. This is the expected behavior from both the static and dynamic policies. The change can be subtle for the channels less that 6.4MHz wide since those have less capacity to begin with, and the speed difference between profiles is a tighter distribution.

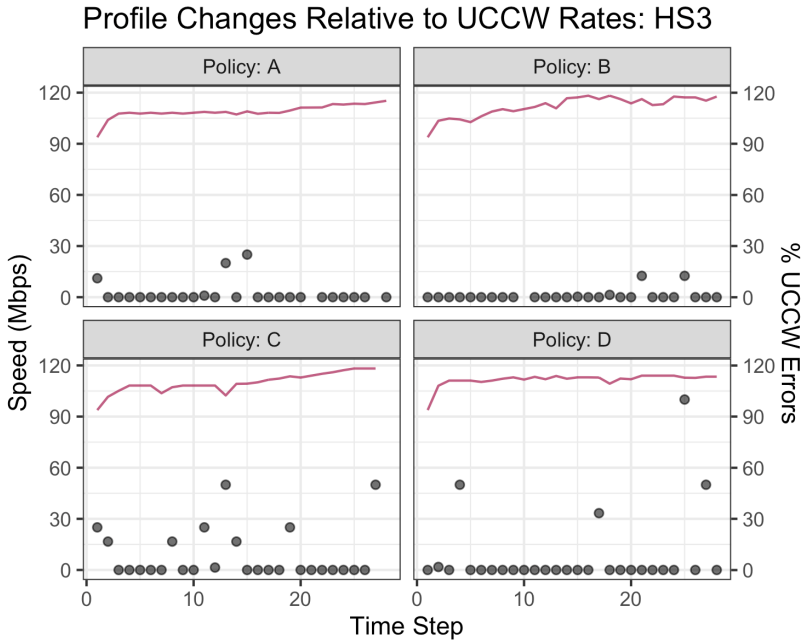

Figure 15 - Changes in Profiles with UCCW Rates on RPD HS3

As shown by policies A, B, and C, as UCCW rates improved, the policies continued to upgrade the profile configurations for more capacity. Observing the trends both with and without noise demonstrates the RL policies' ability to make directionally accurate decisions.

5.3.2.2. Latency to Optimal Profiles from Baseline

The dynamic policies took varying paths on the four-channel systems from the baseline profiles. Policy D took the most efficient route of the policies, reaching a steady state on optimal profiles in an average of two steps (actual = 1.7). Similarly, policy A averaged two steps, however, took a slightly less efficient route. Also notable, as policy D reached the optimal profile, a channel began fluctuating between the optimal profile and a two-step downgrade. This is a flaw within the policy. The same fluctuation activity is observed on policies B and C – both of which had more difficulty achieving and maintaining optimal profiles.

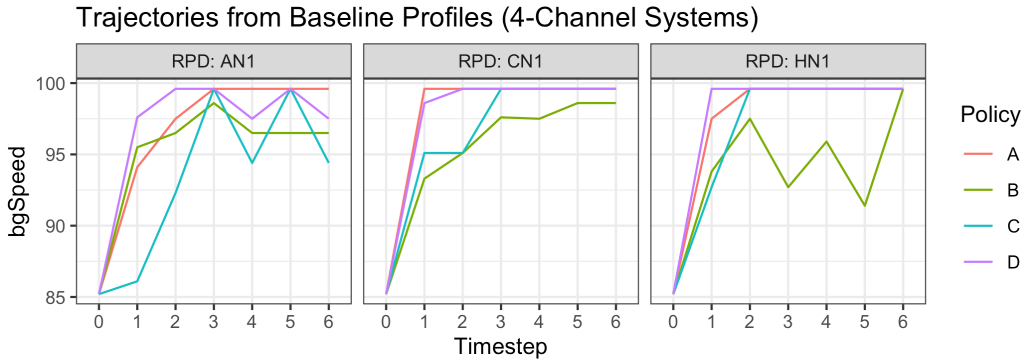

Figure 16 - 4-Channel Upgrade Trajectory from Baseline (Dynamic Policies)

The six-channel systems (AS2, HS3) had a more difficult path toward achieving optimal configurations. With the impairments on HS3, reaching the optimal profiles is not expected; however, reaching the best profile available is expected. The consistency between policies on HS3 indicates that impairments had a strong influence in keeping the profiles sub-optimal. The channels were influenced to use lower profiles to reduce the poor telemetry responses from the network in reaction to the chosen profiles. None of the policies on AS2 achieved optimal profiles, but the speeds were higher, indicating clearer telemetry. Policy D settled early after two timesteps, but at a sub-optimal overall speed. This indicates that the wrong action was valued highest for a particular state. The policy needs to explore different actions to overcome this.

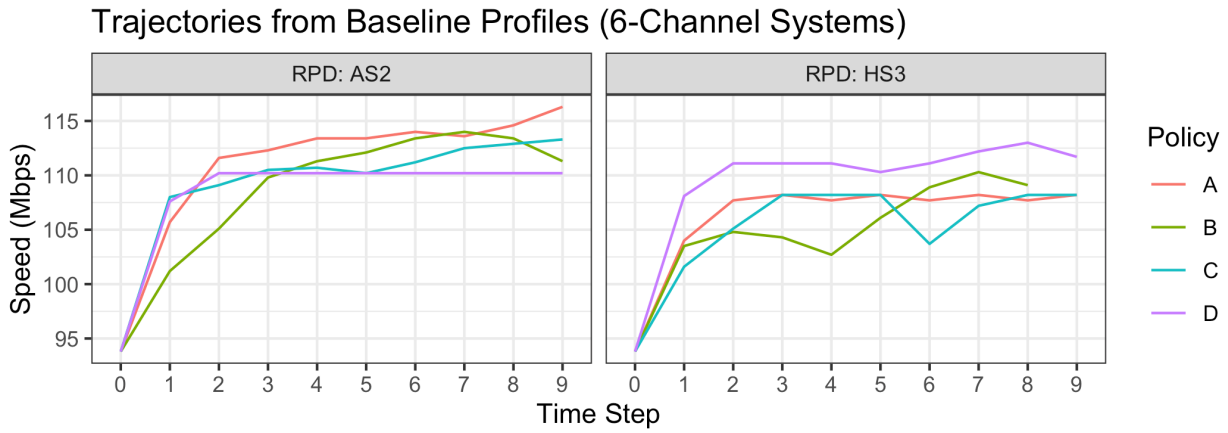

Figure 17 - 6-Channel Upgrade Trajectories from Baseline (Dynamic Policies)

5.3.2.3. Summary Statistics

Over 25 iterations, the dynamic policies achieved varying average bonding group speeds, as shown in Table 5 as an average speed per bonding group.

Table 5 - Dynamic Policy Raw Profile Speed Totals

Policy	Average per Bonding Group
A	104.0 Mbps
B	99.5 Mbps
C	103.2 Mbps
D	103.7 Mbps

Despite none of the dynamic policies achieving the 104.3 Mbps average, the measurement from the static policy, each of the dynamic policies experienced adverse UCCW rates that prevented the policies from upgrading into optimal profiles and remaining there. Yet, Policy A fell just short of attaining the static policy's benchmark.

Figure 18 has the breakdown of profile usage by profile type as it pertains to contributions to the overall speed values. Viewing the results by profile type is useful in understanding how the profiles were utilized across the iterations.

Notable in the profile density plot is the use of optimal profiles for policies A, C, and D in relation to the other profile types. Policies A and D exhibit the best consistency of optimal configurations.

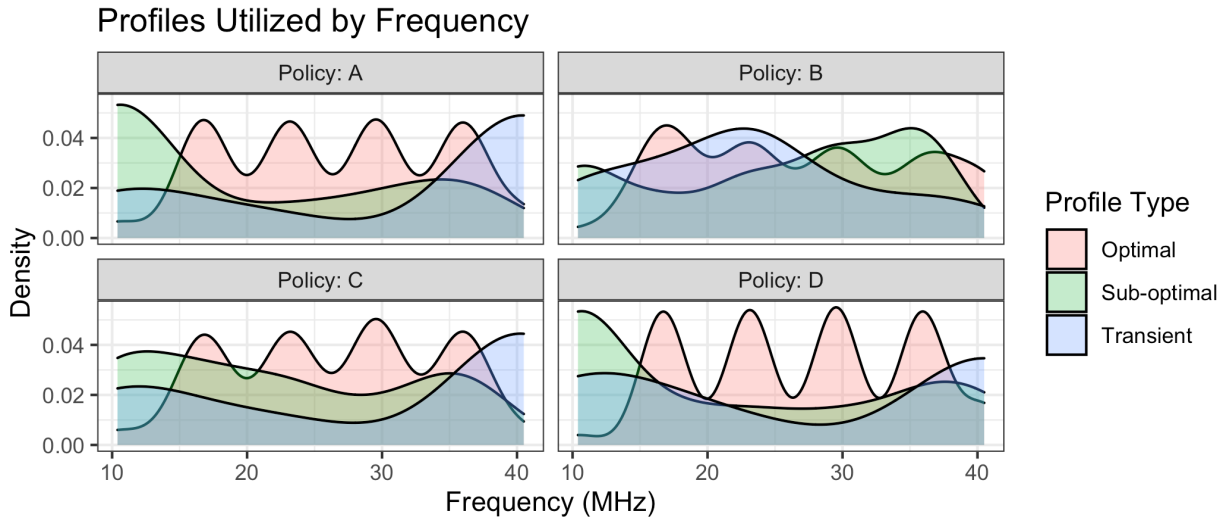

Figure 18 - Profile Type Utilization by Frequency (MHz) for Dynamic Policies

5.3.3. Summary Evaluation

Overall, policy A outperformed the other dynamic policies and will be the subject of a final extended study with respect to the static policy. Both policies ran for a total of 40 timesteps to provide additional insight into whether an RL based policy is definitively capable of being more performant than the static policy. Figure 19 shows the profile speed results comparing the two policies together over time.

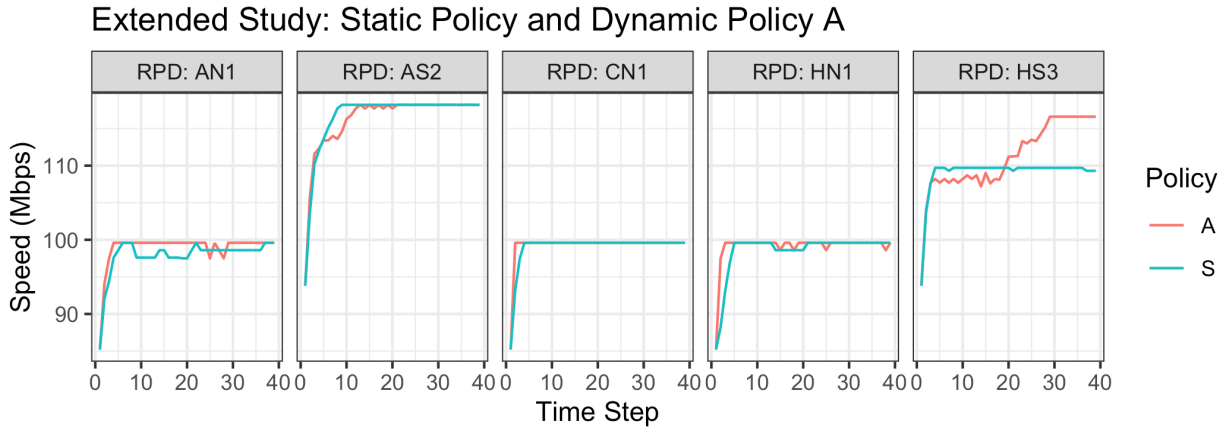

Figure 19 - Profile Speed over Time, Static vs Dynamic

A notable difference is on RPD HS3 when dynamic policy A runs were no longer encountering poor UCCW rates and saw a total increase of approximately 10Mbps from step 20 to step 30. The policy maintained that speed approaching the end of the study. The discrepancy of speed between the two polices is due to a channel reporting erroneous telemetry in the static policy iterations. It is unknown whether the static policy would have encountered poor telemetry or not as it upgraded, therefore it cannot be definitively proven by profile speed alone which is the better policy. Policy A increased to an average of 104.2 Mbps per bonding group of raw profile speed, while the static policy maintained an average of 104.6 Mbps. The high UCCW rates incurred on the systems during the trial runs for policy A explains much of the difference.

One other interesting point related to RL specifically occurred on RPD AS2. Figure 20 shows a zoomed in view of the change that took place. As policy A reached the optimal profile speed, a single channel had learned the highest valued action in that state was to downgrade by one step. Over four total cycles of fluctuating between the optimal profile and the sub-optimal one, the policy was getting updated by being penalized for moving down and rewarded positively by moving up. After those four cycles, the best action from the optimal profile for that channel changed to the action of remaining in the optimal profile under good telemetry conditions.

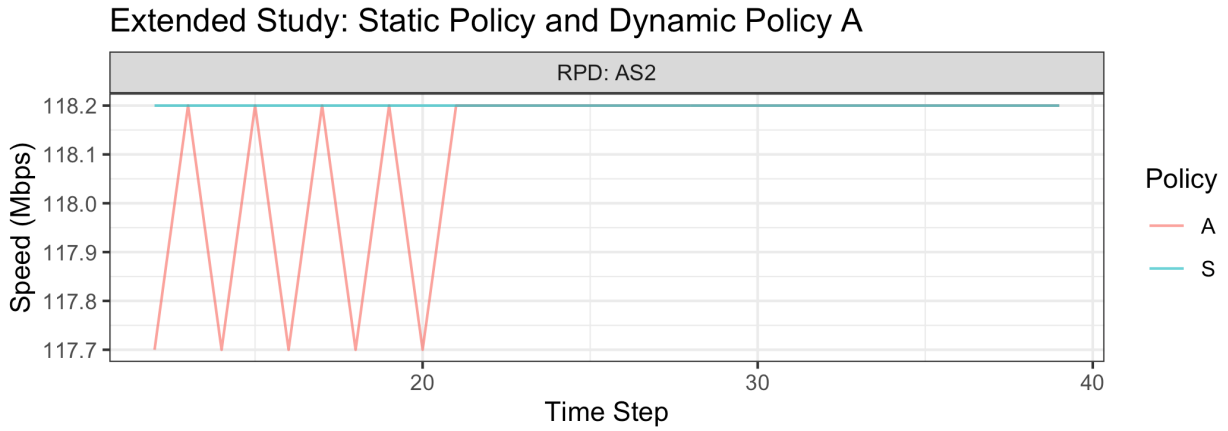

Figure 20 - Dynamic Policy Learning Better Action

Optimal profile configurations on policy A trials account for more of the overall aggregated speed than for the static policy. The static policy spent less time on transient profiles, however that is due to the differences in telemetry between the policy trials. Sub-optimal profiles – those that are between the optimal and transient profiles – accounted for almost twice as much of the total speed on the static policy than on dynamic policy A.

Table 6 - Profile Speed Metrics, Static vs Dynamic

	Policy A		Static Policy	
Profile Type	**% of Total Speed**	**% Profile Occurrences**	**% of Total Speed**	**% Profile Occurrences**
Optimal	91.67%	84.69%	88.95%	85.15%
Sub-optimal	5.11%	5.78%	8.83%	8.44%
Transient	2.42%	6.8%	1.37%	2.24%
Below QAM-64	0.79%	2.72%	0.84%	4.17%

Both policies have proven to behave very similarly in the interest of managing a US PMA system. Dynamic policy A only slightly edges out the static policy based on the following criteria:

- Policy A achieved optimal profiles on the four-channel systems from the baseline in an average of two steps, whereas the static policy took an average of four steps.
- Policy A utilized the optimal profile configurations more often than the static policy.
- Static policy achieved higher overall profile speed over the 40 iterations, with the caveat that the RPDs using policy A experienced more noise from the lab.

A key advantage the dynamic policy has over the static policy is the ability to learn on the fly. These policies are capable of correcting for changing conditions and adapting to what is normal for each RPD

system. As was demonstrated, policy A changed behaviors in the middle of the trial toward the more optimal action – this after experiencing several timesteps with poor telemetry.

While policy A, at a minimum, matches the behavior of the static policy, it also introduces the ability to build customized policies for individual RPD or CMTSs. Geography, weather, ingress, and other external factors impact network service to varying degrees – making a one-size fits all policy advantageous for many systems, but not applicable to all. Changes in one policy will not affect all systems, just the system for which it manages.

Balancing the many permutations of telemetry and configuration values using RL states removes complexities involved with tuning thresholds and applying the conditional logic in the algorithm for all thresholds.

5.4. Opportunities for Enhancement / Potential Future Steps

The proof of concept described in this paper is a promising step toward building more intelligent PMA systems. Below is a list of opportunities to improve the current policy building process and suggestions for future steps that would move this work forward.

- Training a dynamic policy from scratch – given enough time and resources, a policy trained from scratch would not be influenced by the initial historical data used in this POC.
- Synchronize noise settings from the lab such that each RPD and policy experiences the same sequence of impairments. Measuring responses of devices at each transition would provide a clearer picture of policy behavior differences.
- Improve the policy training process using n-step TD prediction methods, whereby more steps in the sequence and the rewards from those steps is used to estimate state-action values. This POC used the current state/action and next state/action to calculate values. Additional states and actions in the sequence can be bootstrapped for learning.
- Adaptation for scalability – architect solution capable of managing tens of thousands of RPDs. This may sound daunting at initial glance, however, the state-action space to maintain is significantly smaller than the collective set of possible states. In this study, .007% of the 243K possible states were encountered.

6. Conclusion

An upstream PMA system operating through a static policy is a proven effective strategy for configuring D3.0 channels across an entire network. By taking a cautious approach, the single policy caters more to the adversely impacted RPDs and CMTSs as the lowest common denominator when establishing thresholds that need to apply to a wide range of devices and conditions.

To improve profile configuration management, RPDs and CMTSs would benefit from a policy that best suits the individual operational environments. In fact, through the current implementation, a primary static policy manages the majority of Comcast's footprint, and a secondary policy manages a small set of devices with special requirements. Following a path of creating multiple static policies that manage different sets of devices would become difficult to manage.

One option for establishing self-managed dynamic policies is to apply an RL-based decision-making process that updates in real-time and is flexible enough to tune for either groups of devices or individual devices. The finer-grained management leads to confidently bolder profile transitions to reach steady state operations in less time than the static implementation. While both policies performed similarly in the trial,

the advantages a PMA system gets with an RL implementation may make the RL approach more appealing for large networks.

Abbreviations

CCW	correctable codewords
CM	cable modem
CMTS	cable modem termination system
CPE	customer premise equipment
DAAS	Distributed Access Architecture Switch
D3.0	DOCSIS 3.0
D3.1	DOCSIS 3.1
DOCSIS	Data Over Cable Service Interface Specification
dB	decibel
IL	imitation learning
IRL	inverse reinforcement learning
Mbps	megabits per second
MDP	Markov Decision Process
MHz	megahertz
α	alpha
γ	gamma
PHY	physical layer
PMA	Profile Management Application
POC	proof of concept
QAM	quadrature amplitude modulation
RL	reinforcement learning
RPD	Remote PHY Device
SARSA	state, action, reward, state, action
SCPI	Standard Commands for Programmable Instrumentation
SNR	signal to noise ratio
TD	temporal difference
UCCW	uncorrectable codewords
US	upstream
vCORE	core voltage
VLAN	virtual local area network

Bibliography & References

1. *Full Scale Deployment of PMA.* Harb, M, et al. s.l. : NCTA technical paper, 2020.

2. *Reinforcement Learning: As Introduction.* Sutton, R. S., Bach, F., & Barto, A. G.. s.l. : MIT Press Ltd, 2018.

3. *Algorithms for Inverse Reinforcement Learning*. Ng, A, Russell, S. Berkeley : s.n., 2000.

4. *A Reinforcement Learning Framework for Optimizing Throughput in DOCSIS Networks. Dugan, K., Harb, M., Rice, D.,* In Workshop on Flexible Networks Artificial Intelligence Supported Network Flexibility and Agility (FlexNets'21), August 27, 2021, Virtual Event, USA. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3472735.3473389

Additional References

ANSI C63.5-2006: *American National Standard Electromagnetic Compatibility–Radiated Emission Measurements in Electromagnetic Interference (EMI) Control–Calibration of Antennas (9 kHz to 40 GHz)*; Institute of Electrical and Electronics Engineers

The ARRL Antenna Book, 20th Ed.; American Radio Relay League

Code of Federal Regulations, Title 47, Part 76

Reflections: Transmission Lines and Antennas, M. Walter Maxwell; American Radio Relay League

