PNNL-38909

From Modular ADMS to Plug-and-Play Ops

Distribution Grid Operations with Platform-Level Orchestration to Enable Ambitious App Hosting

May 2026

Andrew P. Reiman
Monish Mukherjee

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

1.0 Introduction

The core function of the distribution grid is to provide electricity to consumers affordably, reliably, and securely. In pursuing these core objectives, distribution utilities are accountable to customers, regulators, and in some cases, shareholders. Other third parties such as aggregators and microgrids can also have a stake in the smooth operation of the grid. Each of these stakeholders has economic, business, and/or governance objectives that inform their expectations of the distribution grid. This multi-objective, multi-stakeholder environment creates tension that must be reconciled to successfully design and operate the distribution grid.

Innovative companies are competing to bring high-tech solutions to electric utilities and their customers that address each of these objectives. Many developers of advanced distribution management systems (ADMS) and distributed energy resource management systems (DERMS) have adopted a modular architecture that allows grid operators to select functions and features according to their individual system needs. A modular platform also allows the solution provider to develop and integrate specific new product modules; however, the need to pursue multiple objectives with a fixed set of controllable devices makes integration expensive whether it is done at the product development stage or the deployment stage. This cost creates a significant barrier to adoption and can lengthen the product to market time of new solutions.

To fundamentally address the complexity of system integration for distribution grid operations, the U.S. Department of Energy Office of Electricity has funded the GridAPPS-D project at PNNL, which streamlines integration by contributing to standards development, defining system architecture, applying advanced mathematics, and developing open-source software to demonstrate the concept of an open data-integration platform for distribution operations. The open data-integration platform concept enables system operators and solution providers to deploy ambitious, best-of-breed applications (or apps) without continually reengineering for integration.

Ambitious apps developed by different solution providers will inevitably attempt to achieve different control objectives with the same set of controllable devices. If the open platform itself can resolve these conflicts in a way that achieves the best available outcomes for all apps, doesn’t restrict the ambitious design of apps, and ensures safe and secure operations, apps will be able to plug-and-play with the platform at the same time as other ambitious apps. In this paper, we describe a framework called App Deconfliction that empowers a platform to assign setpoints to controllable devices based on the values preferred by different apps (and even external stakeholder entities like customers or aggregators).

The App Deconfliction framework is compatible with several methods for determining setpoint values. We present two methods based on game theory that provide a subtle built-in incentive structure for developers to adapt their apps to the fact that they will be operating in a moderated multi-app environment and to favor device setpoints that have the most effect on their objectives over those that have the least effect. Our simulation-based demonstrations have shown that game-theory-based deconfliction can lead to a 7% improvement in control space utilization compared to design-based methods.

2.0 Apps, Entities, and Conflict

To provide affordable, reliable, and secure energy to consumers, distribution utilities must manage or maintain at least six operational parameters or states of the grid: 1) power, 2) voltage, 3) power quality (power factor and harmonics), 4) frequency, 5) topology, and 6) stored energy. A non-exhaustive list of controllable assets that impact the operational states is shown in Table 1.

Table 1 Typical Controllable Assets for Distribution Grid Operations

Switches	Capacitors
Voltage regulators	Generation assets
Energy Storage	Demand management

In addition to the assets listed in Table 1, engineering, business, and regulatory factors can also affect the operational state of the grid, but these factors cannot be controlled in an operational context.

In an open data-integration platform, different applications and distributed agents (collectively referred to as apps) can be deployed to pursue different objectives that are often based on quantitative metrics. Several examples of these objective-basis quantities are shown in Table 2.

Table 2 Example Objective-Basis Quantities for Distribution Grid Apps

Customer Experience	**Grid Management**	**Energy Services**[1]
Cost of power	Switching configuration	Energy management
Voltage profile & CVR	Protection	Frequency response
Flicker	Fault location	Voltage management
Harmonics	Voltage limits (ANSI C84.1)	Blackstart
SAIDI & SAIFI	Feeder power factor	Regulation
Power factor profile	Emergency load management	Reserve

The list of quantities in Table 2 is not exhaustive and some quantities overlap. Even with plug-and-play apps, it will be important to make design decisions about which set of apps to deploy. Notably, several of the quantities listed in Table 2 are of interest to entities outside of the distribution utility. Conflict can occur between different apps representing a single entity (like a distribution utility) or between apps representing different entities. (When considering conflict between entities it is important to respect ownership of signals and the fact that entities will not accept external influence without a compelling reason such as compensation, incentive, or regulatory requirement.)

Consider the following set of apps that may be of interest to a distribution network operator and other grid stakeholder entities:

- Peak load management: this app attempts to use distributed generation (like photovoltaics and diesel generators) as well as storage reduce peak load of distribution network and/or remain below transmission supply price breaks.

---

[1] Energy Services Interface https://www.osti.gov/servlets/purl/1992370/

- Conservation Voltage Reduction (CVR): This app uses assets like capacitors, voltage regulators, and reactive power to keep the system voltage profile close to the lower operating limit.
- Feeder power factor optimization: This app uses capacitors and reactive power injections to keep the power factor at the feeder head close to unity to meet transmission system obligations.
- Backup power: this app attempts to preserve energy (in form of diesel fuel and stored energy) to provide backup power in case of an outage.
- Energy arbitrage: this app attempts to maximize the value of energy storage assets by buying low and selling high.

This set of apps produces clear tension around at least four of the six operational states (power, voltage, power quality, and energy) as well as several classes of controllable devices. To deploy all of these apps together would require either careful and restrictive integration design or a system-level capability to actively deconflict apps.

3.0 Section 3: App Deconfliction

Conflicts between device-controlling apps must be avoided or resolved to avoid a race condition on device setpoints where the behavior of the system is determined by the most recent app to produce a setpoint. Today, platform developers and system integrators can apply design approaches to preempt race conditions. Options include assigning priority to functions, apps, or agents on a per-device basis, according to a schedule, and/or based on the state or condition of the system. Collectively, we refer to these options as designed noninterference.

While designed noninterference can be considered a branch of App Deconfliction, the framework also supports methods that can dynamically consider input from all apps. We refer to these dynamic methods as active deconfliction. Active deconfliction is implemented by routing setpoints produced by all apps through a system-level function that resolves conflicts and determines a single value for each setpoint. A comparison of high-level approaches to designed noninterference and active deconfliction is shown in Table 3.

Table 3 Designed Noninterference and Active Approaches for App Deconfliction

Approach	Description
Device-Based Designed Noninterference	Each app is assigned priority to control specific devices. Priority may be app-exclusive or hierarchical.
Time-Based Designed Noninterference	App priority to control of each device is determined according to a schedule.
Event-Based Designed Noninterference	Device control depends on system modalities and may change in response to events.
Multi-Objective Optimization (MOO)	App objectives are combined and solved using a centralized optimizer or distributed optimization.

Game Theoretic Active Deconfliction	A mediator with a predefined action set incentivizes apps to prioritize high-leverage setpoints over others.

3.1 Designed Noninterference

Designed noninterference approaches give integrated system architects and integrators tools to preempt app conflict. But these solutions share a common limiting feature: primary control authority for each device is assigned to a single app at a time. This makes it difficult or impossible to achieve balance between the various objectives of competing apps.

3.2 Multi-Objective Optimization

One way to directly consider the objectives of all apps is to construct a composite system-level objective from all relevant app objectives, then solve using multi-objective optimization (MOO). This would require the system access all app objectives and understand how to combine them. It also transfers the burden of computing setpoints to the system level—individual apps are effectively replaced by a single app performing MOO. In utility operations, there are two major drawbacks to this approach. First, not all grid operations objectives are readily combined without using heuristic weighting factors or normalization bases. Second, single-app MOO defeats the advantages of modular or open platform architectures.

3.3 Active App Deconfliction

To address the limitations of MOO while still balancing all operating objectives, we have developed an approach that separates the deconfliction process both from apps and from the system designer role. Under Active Deconfliction, a deconfliction pipeline processes setpoints provided by all apps to determine a single value for each controllable device setpoint. The deconfliction pipeline architecture is shown in Figure 1 with inputs from internal apps and external entities (discussed more below).

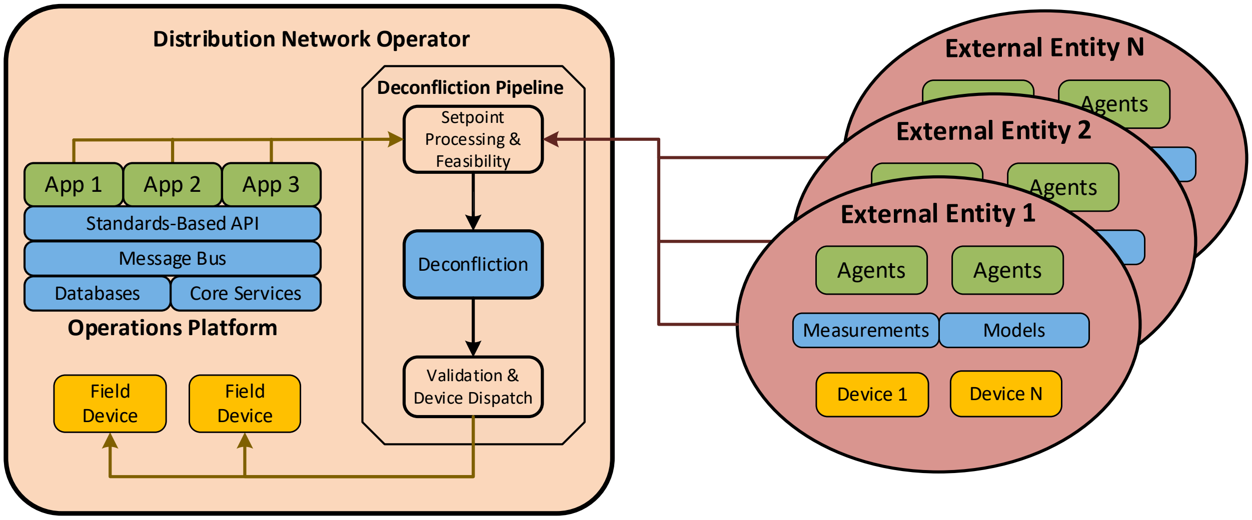

Figure 1 App Deconfliction Pipeline Architecture

The deconfliction pipeline includes three stages:

- A Setpoint Processing & Feasibility stage, which interprets setpoints sent by apps to maintain the best available understanding of the continuous intent of apps and ensures that all setpoints are feasible;
- A Deconfliction stage, which implements a specific deconfliction solution and identifies a single value for each controllable device setpoint; and,
- A Validation & Device Dispatch stage, which ensures that deconfliction solutions satisfy system safety, security, and operational requirements before dispatching devices as needed to track the deconflicted setpoints identified in the Deconfliction stage.

3.4 Combination Approaches for Practical App Deconfliction

Designed noninterference concepts can be combined with active deconfliction concepts. In the design stage, rules and heuristics can be built to introduce static, modal and/or operator-determined constraints for active deconfliction. For example, active deconfliction can be designed to be disabled under emergency conditions or upon operator intervention. Alternatively, or in addition, static or operator-tunable weights can be incorporated into a modified weighted direction game. This flexibility in the app deconfliction framework ensures that grid operators can maintain discretionary control over the deconfliction process and the grid at large.

4.0 Game Theoretic Solutions for App Deconfliction

Under Active deconfliction, a range of solutions can be employed during the deconfliction stage of the deconfliction pipeline. Solutions must produce a single value for each controllable device setpoint given input from multiple apps; they should also do this in a way that enables each app to be as productive as possible in the multi-app context. We have identified the following desired criteria for deconfliction solutions:

- Must produce a single value for each controllable device setpoint,
- Should allow all apps to influence setpoint outcomes,
- Should enable apps to be as productive as possible collectively,
- Should not place unnecessary design constraints on apps,
- Should incentivize apps to demonstrate cooperative behavior,
- Should not marginalize apps that simply state their desired outcome; and,
- Should not introduce undue computational burden to apps.

Game theory provides a framework can satisfy all of these criteria: the deconfliction stage can host a mediator with a predetermined process for converting app-generated target setpoint values into controllable device setpoint values (in game theory terminology, the mediator is a leader-player with an open strategy). If the mediator's strategy is designed carefully, each app will be incentivized to play the game in a way that prioritizes outcomes that are most important (positively or negatively) to its bottom line and deprioritizes outcomes that it is less sensitive to.

We will provide a high-level description of two deconfliction games:

Weighted Consensus Game: In this game, the mediator iteratively computes a weighted centroid of app-supplied device setpoint vectors. Weighting factors are computed by the mediator based on the flexibility demonstrated by each app. In the initial iteration, each app supplies its “greedy” setpoint vector and the mediator computes an initial centroid. In subsequent iterations, each app competes for a larger weighting factor by accepting the consensus value of setpoints that have a minimal effect on the app objective. In each iteration, the mediator updates weighting and computes a new weighted centroid. This game empowers apps to compete for the best possible outcome given the multi-app environment and even rewards apps that can accurately predict the behavior of other apps; however, this game has drawbacks: 1) the game does not meet the las desired criterion for deconfliction solutions—to pursue optimal outcomes, apps must solve a complicated nonlinear objective (with complexity that increases with detailed modeling of other apps’ behavior), 2) the game takes many iterations to converge (asymptotic convergence).

Weighted Direction Game: In this game, the mediator constructs a linear global optimization problem based on the implied success criteria of each app encoded in the controllable device setpoints provided by the app. Specifically, the mediator constructs a maximum global utility problem with objective equal to a sum of setpoint vectors provided by each app normalized to have uniform length. Baring information about the behavior of other apps, each app is incentivized to hedge proportionally towards outcomes that it strongly favors and away from outcomes that it strongly disfavors. Because apps know that the mediator will construct a linear program, apps can take advantage of the fact that the mediator’s solution will lie at a vertex of the feasible space. Apps can then internally compute their objective value at each vertex of the feasible space and construct a setpoint vector that is the sum of vertex vectors weighted by the corresponding objective value. This game still incentivizes apps to cooperate, is more computationally tractable to participating apps, and avoids time and computational cost of iterating; however, it may be possible for apps to gain an unfair advantage in this game by sensing the priorities of other apps and adjusting setpoints accordingly.

5.0 Demonstration of Active Deconfliction

To illustrate how operational conflicts can arise, we will describe an example with three apps with the following objectives: minimum cost of energy (App1), maximum reserve energy (App2) and optimal CVR (App3). Figure 3 shows the preferred controllable device setpoint values for each app and illustrates the inherent conflict between these apps. App deconfliction can be used to ensure safe, reliable, and efficient operation of the distribution grid. We will provide an overview of how app deconfliction can address key challenges that arise in modular platforms.

Resolving Conflict: The fundamental role of app deconfliction is to prevent conflicting control actions from different apps from affecting device operation. Without a deconfliction mechanism, such conflicts may result in unintended system behavior including voltage excursions, equipment stress, or degraded system performance. Building on the example described above, Figure 2 illustrates how app-deconfliction can identify and resolve such operational conflicts during execution.

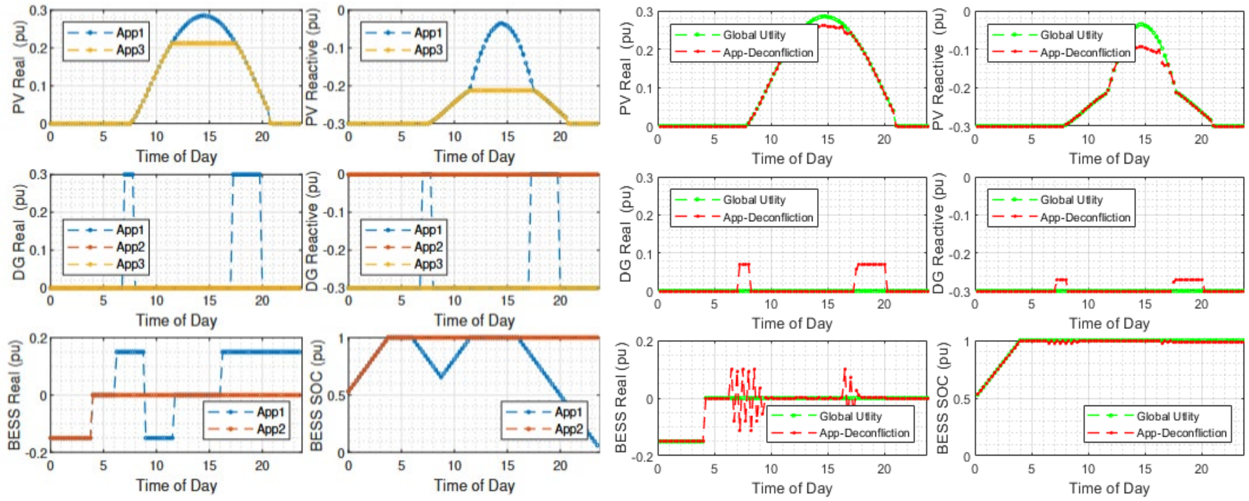

Figure 2 Conflicting setpoints produced by three apps operating exclusively (left) and deconflicted setpoint values after implementing game theoretic app-deconfliction (right).

Unlocking Enhanced System Performance: Without deconfliction, apps with overlapping spans of control can undermine the benefits of other apps, thereby creating operational inefficiency. Designed noninterference restricts the access of certain apps to certain devices and constrains the effectiveness of all apps. However, well executed active deconfliction enhances system performance by dynamically balancing app objectives across all devices and empowers each app to pursue the best outcomes available in a multi-app context.

Figure 2 shows deconflicted setpoints along with the setpoints that achieve global utility based on the three apps in our example. This result illustrates the ability of active app deconfliction to achieve near-optimal trade-offs between the objectives of multiple apps. Table 4 shows a comparison of the performance of DN and Active Deconfliction. Even though both time-based and device-based DN mechanisms were carefully designed to perform well in this scenario, the active deconfliction solution (implemented here with the weighed directed game solution and no scenario-specific design or tuning) outperforms the best-case DN approach by 7% in terms of average app success. This is further illustrated by the bar chart in Figure 3, which shows the intrinsic tension between the apps when operated exclusively and the relative performance of the deconfliction solutions.

Table 4 Percentage Performance of Designed Noninterference and Active Approaches for App Deconfliction

Scenarios	**Cost Savings**	**Energy - Reserve**	**Voltage Reduction**	**Average**	**Range**
Affordability Exclusive	100	86.5	54.7	80.4	45.3
Reserve Exclusive	88.9	100	57.4	82.1	42.6

CVR Exclusive	77	87.4	100	88.1	23
DN: Time	89.1	83.9	68.4	80.5	20.7
DN: Device	89.5	94.5	83.2	89.1	11.4
Active Deconfliction	88.9	100	97.2	95.4	11.1

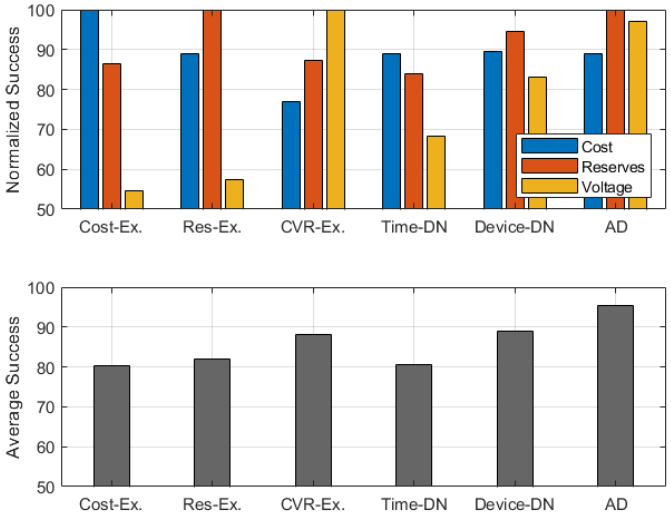

Figure 3 Performance of Active Deconfliction compared to Designed Noninterference

Reliability and Security: Uncoordinated and unsupervised application behavior can compromise system reliability and safety. A modular operations platform, especially if it is open to apps from multiple vendors introduces a security vulnerability, wherein a malfunctioning app or an app with a malicious strategy could impact a large number of devices and create grid instability. App deconfliction can track and detect malicious behaviors and thereby build robustness against malicious app strategies. Figure 4 shows the impact of the presence of a malicious app on system and demonstrates how deconfliction can improve the system performance by detecting and isolating influence of malicious apps.

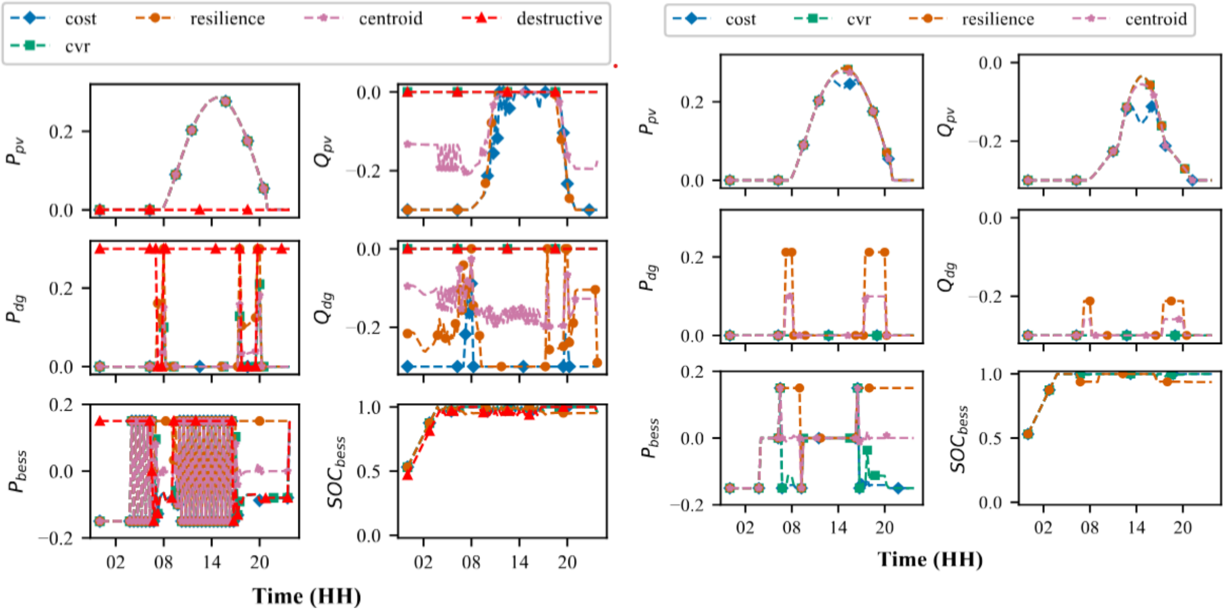

Figure 4 Device control setpoints from each app & deconflicted solution in presence of a malicious app (left) and the deconfliction solutions after detecting and isolating (left) and the case with a malicious app (right).

Scalability and integration: from an organizational perspective, application deconfliction reduces workload of system operators. As utilities adopt more intelligent and autonomous applications, operators face the risk of receiving conflicting recommendations that increase the decision-making burden. From the solution-provider perspective, platforms with modules and features that fit the unique needs of utilities typically require specialized integration that increases costs and extends the product development cycle. Integrating new modules and features into an already operational platform can be even more costly. A structured app-deconfliction mechanism simplifies the customization and integration process, streamlines decision-making, and strengthens cyber-resilience by detecting anomalous or unauthorized commands. In sum. app deconfliction harmonizes the growing ecosystem of digital tools in distribution operations.

6.0 Multi-Entity Deconfliction

As shown in Figure 1, the App Deconfliction framework supports consideration target setpoint values provided by external entities. In this multi-entity deconfliction paradigm, which can be implemented by a distribution utility or any other grid stakeholder, controllable-signal ownership must be respected and entities are not compelled to respect setpoint values provided by external entities. Rather, entities determine whether and how to respond to external inputs based on factors like negotiated exchange, compensation, incentive, leverage, or regulatory framework. Under multi-entity deconfliction signal owners can choose to consider input from external entities in addition to internal factors in determining values for controllable device setpoints.

6.1 Multi-Entity Deconfliction Example

The dynamic operating envelope paradigm can be used to understand multi-entity deconfliction. When aggregated or independently operated distributed generation are deployed at levels that can create congestion on the grid, limitations must be imposed by the grid operator to either limit the deployment of additional assets or dynamically regulate the operation of these assets. A dynamic operating envelope is one such regulation mechanism whereby a grid operator determines location-specific (e.g., per-feeder) limits on distributed generation at specified operational intervals. In Multi-entity deconfliction terms, an aggregator can be said to own the generator dispatch signals and a grid operator can be said to own an emergency asset disconnection signal. The grid operator can generate a dynamic operating envelope signal and transmit it to the aggregator. The aggregator can perform multi-entity deconfliction understanding that the grid operator controls emergency asset disconnection signals and choose whether to adopt the dynamic operating envelope setpoint as a constraint on its generation dispatch signals. Meanwhile, the grid operator will determine whether to actuate emergency dispatch signals as needed to maintain safe and secure operation within regulatory limits. The optimal strategy for the aggregator will be to adopt the dynamic operating envelope.

6.2 Additional Considerations for Multi-Entity Deconfliction

Multi-entity deconfliction is a process whereby each entity can make decisions that may be contingent on decisions made by another party. It describes how two entities can systematically achieve a structured consensus although it does not specify how agreements should be brokered or enforced. Multi-entity deconfliction can also describe how non-monetary signals like data and model information can be exchanged to complement a transactive energy framework.

7.0 Section 7: The Path Forward

The GridAPPS-D team continues to pursue more streamlined integration of advanced technology into distribution grid operations. Ongoing efforts include applying mechanism design (a branch of game theory focused on formally designing games to achieve desired outcomes through optimal player strategies) to improve deconfliction solutions, extending game-theory-based solutions for single-entity deconfliction to multiple entities, and examining how selective exchange of model information and data can supplement and enhance multi-entity deconfliction.

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov