Dynamic Decision Making for Demand Response through Adaptive Event Stream Monitoring

Gururaghav Raman, Jimmy Chih-Hsien Peng
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
Email: gururaghav.raman@u.nus.edu, jpeng@nus.edu.sg

Bo Zhao, Matthias Weidlich
Institut für Informatik
Humboldt-Universität zu Berlin, Germany
Email: {bo.zhao, matthias.weidlich}@hu-berlin.de

***Abstract*—The traditional approaches for implementing event-based Demand Response (DR) have been static, and do not involve feedback to the consumers regardless of their performance during the DR event. This may however lead to an incomplete system-wide response, thereby forcing the utility to employ direct load control to achieve the required response, or buy additional generation reserve from the spot market. To mitigate this inefficiency, this paper proposes closing the loop through an incentive control for residential DR participants enabled using event stream monitoring. By realizing the latter in an adaptive and distributed manner, the data communication and computation overhead involved in the decision making process is reduced. With simple assumptions, it is demonstrated that methods for event stream processing can ensure that the scheduled DR is achieved completely. Therefore, the proposed implementation allows for scalable, effective, privacy-preserving, and robust implementation of incentive-based residential DR that ensures full overall compliance to the DR task.**

***Index Terms*—adaptive event stream monitoring; demand response; incentive-based; smart grid.**

I. Introduction

Residential demand response (DR) is an essential part of a smart grid, and refers to the actions taken by the consumers to alter their consumption as requested by the utility. DR implementations may in general be price-based, event (incentive)-based, or direct load control (DLC). The first implements a time-varying tariff in order to shift consumption away from the peak demand period, whereas the second offers an incentive to the consumers to reduce their consumption during a specified event period. The third, as the name suggests, involves the utility directly controlling the end-user's loads such as air conditioning or lighting as required. While price-based DR is not dispatchable, event-based and DLC responses could be obtained during an exact interval specified by the utility. These may therefore be more useful to the grid operator to alleviate stress on the grid during the peak demand hours. However, among these, DLC programs have historically experienced low adoption rates owing to the residents' reluctance to losing control of their flexible loads to the utility [1]. Therefore, incentive-based DR implementations are expected to have a high penetration in the future distribution grid.

The implementation of event-based DR, especially in the residential sector, has traditionally been open-loop, as illustrated in Fig. 1. The utility generates a DR request that is communicated to the DR participants. Based on an expected

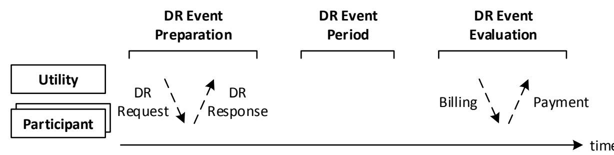

Fig. 1. Traditional approach to event-based DR.

response from these consumers, the utility schedules other resources such as conventional or distributed generation or storage to achieve the appropriate reserve levels. In this scenario, no further communication or feedback is sent to the consumer during the actual DR event period. The success of the DR event, and the conformance of the consumers is only assessed in retrospect, after the fact, for billing purposes. This static approach has drawbacks because it lacks flexibility: first, in terms of the granularity at which trade-offs (offers to consumers vs. coping with increased energy consumption) are managed. There is only one DR request issued for the whole network, which is moreover before the time frame in which the savings shall materialize. Second, there is no flexibility or means for the utility to manage resources if the overall system-wide response falls short of the expected value, forcing it to turn to other resources such as the DLC flexibility provided by the commercial and industrial sector. Clearly, this is not economically optimal due to the under-utilization of the residential demand flexibility.

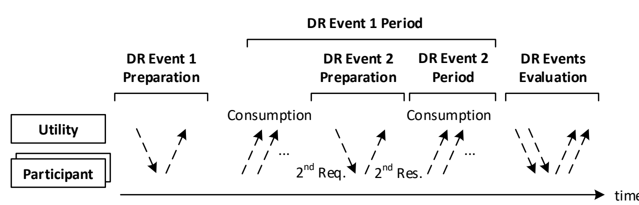

Fig. 2. Proposed approach to event-based DR including a feedback loop.

This paper therefore argues for a dynamic approach to incentive control in an event-based DR implementation, as outlined in Fig. 2. This has at its core, a feedback loop from the utility to the consumer that enables the utility to announce new incentives—a second DR event—to a subset of the participants within the time frame of the first DR event as it unfolds. Any dynamic control approach must meet the following requirements:

- *Distributed processing:* the computation underlying decision making needs to be distributed to achieve scalability

to large-scale networks, to ensure consumer privacy, and to avoid a single-point of failure [2], thereby achieving fault tolerance in decision making,
- *Traceable processing:* the decision making shall be explicit, not hidden in black-box models in order to enable manual monitoring, and finally
- *Online processing:* to enable dynamic decision making, data on the behavior of consumers needs to be processed immediately, with low-latency.

With these objectives, the contributions of this paper are summarized as follows. It proposes a model of dynamic decision making for event-based DR, which ensures that the full response is obtained from a group of residential consumers. Also provided is a design of the technical infrastructure that achieves this by using a feedback from the utility that adjusts the incentive to a subset of the consumers based on adaptive stream monitoring of the real time behavior of the entire group. By distributing the monitoring functionality, communication and computation costs are reduced to achieve scalability. Results illustrate that by employing distributed event stream processing, an order of magnitude of both computation and communication costs can be reduced.

II. Background

This section presents essential concepts regarding residential DR (Section II-A), as well as the basic notions of event stream processing (Section II-B).

A. Residential Demand Response

This subsection details the DR implementation (the static case with no feedback), as well as the simulation procedure employed in this paper for generating synthetic residential load profiles with and without DR events.

1) Event-based DR implementation: In event-based implementations, as the name suggests, the response from the consumer is triggered by an explicit signal from the utility that specifies the time of the event, the energy reduction requested, and the incentive for the same response. The following happen sequentially if a consumer response is required in the distribution system.

First the utility generates a DR event for a set of participants (along with their volume of response, and the corresponding timeframe $[t_{start}, t_{end}]$). The specifications of the DR task are conveyed to the consumers usually using either a dedicated communication network, or an internet-based infrastructure. Second, the consumer respond to the event by either accepting, or declining the DR request [3]. The consumer responses (viz. 'I participate!', or 'I decline to participate!') are collected by the utility, which then estimates the available system flexibility. Third, during the DR event period, the consumers who accept the event take steps to change their appliance usage patterns to meet the DR event specifications. Finally, after the event ends, the utility estimates the participants' contributions during the event, and compensates them for the same. The estimation of the participants' response during the event requires the calculation of the consumer baseline load (see [4] for an overview). Note that a consumer who participates must provide the full volume of response to receive the incentive. Given the event-time real power consumed by a consumer k, $P_{k,t}$, and the corresponding baseline $B_{k,t}$, this is represented as

$$\sum_{t=t_{start}}^{t_{end}} (B_{k,t} - P_{k,t}) \geq \text{Requested reduction.} \tag{1}$$

2) Simulating residential load profiles: The demand profile for each home is generated using a bottom-up model wherein the probabilities of usage $p_{t,i}$ of each appliance i are assumed to be known. These are obtained from time-use surveys such as [5], the data from which is used in our study. The state of each appliance $x_{t,i}$ (1 if on, 0 if off) at time t is determined by generating a random number $rand$:

$$x_{t,i} = \begin{cases} 1, & \text{if } rand \leq p_{t,i}. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The product of the state with the appliance's rating results in the power consumed by that appliance. This procedure is repeated for each time step, and appliance, the sum of whose powers results in the net residential demand. Equation (2) pertains to the case when no DR event is generated. If, however, DR is requested by the utility during the time period $[t_{start}, t_{end}]$, the probabilities of using flexible home appliances (washing machine, dryer, dishwasher) are amended based on the degree of compliance of each consumer $degComp$ to the DR task:

$$p^*_{t,i} = p_{t,i}\ (1 - u \times degComp), \forall t \in [t_{start}, t_{end}], \text{and} \tag{3}$$

$$p^*_{t_r,i} = p_{t_r,i} + \sum_{t=t_{start}}^{t_{end}} (p_{t,i} - p^*_{t,i}). \tag{4}$$

These equations describe the deferment of the use of the flexible appliances away from the DR period, to another randomly chosen time t_r in the rescheduling period. This time t_r is chosen with a probability that is proportional to the original probability of use $p_{t_r,i}$. The parameter $u \in [0, 1]$ refers to the impact of the incentive offered by the utility on the consumer response. Clearly, the higher the consumer compliance and the incentive, the higher is the value of $degComp$, and therefore the higher is the power reduction during the event period.

B. Event Stream Processing

Our approach to dynamic decision making for demand response exploits concepts of event stream processing [6]. Events denote 'occurrences of interest', which may represent low-level measurements (e.g., the current energy at a smart meter) as well as high-level observations (e.g., for a set of smart meters, the accumulated energy is larger than expected). In any case, following [7], events are represented by a relational data model. An *event schema* is a sequence of attributes $A = \langle A_1, \ldots A_n \rangle$, each being of a primitive data type. An instance $e = \langle a_1, \ldots a_n \rangle$ of such a schema is an *event*, with a_k being the value of the respective attribute A_k. As a short-hand, $e.A_k$ denotes the value a_k of attribute A_k of event e.

Attribute:	id	time	houseID	power (kW)
Domain:	$\mathbb{N}$	$\mathbb{N}$	*text*	$\mathbb{R}$
	11	24234982	'H1'	12.20
	12	24235323	'H2'	4.78
	23	24236728	'H1'	12.41

As an example, consider events that measure the power of a house. Such events may be captured as shown above.

An *event stream* of schema A is an infinite sequence $S_A = \langle e_1, e_2, \ldots \rangle$ of events of schema A. Here, the stream order respects the temporal order of events: for events e_j and e_k, $j < k$ implies that $e_j.time \leq e_k.time$.

This paper further adopts a query model that lifts relational queries from static data to the above notion of an event stream [7]. A standard relational query that is evaluated over a set of static data elements follows a `Select-From-Where` structure that defines which attribute values to consider, from which data source, satisfying which condition. In addition, such a query may contain aggregation operators, such as `Sum` or `Count` that are applied over some data elements. If aggregates shall be derived per set of data elements, a `Group By` operator enables the definition of a partitioning based on the values of some attribute before applying the aggregation operator.

Evaluating such queries over streams requires the definition of (i) how to select a set of events (i.e., data elements) from an input stream (i.e., an infinite sequence of data elements), and (ii) how to construct an output stream from the set of events obtained after applying the query operators to the events selected from the input stream. As for the first question, a time-based window may be applied, defined by a size w and a slide s. Given a stream $S_A = \langle e_1, e_2, \ldots \rangle$, a first window selects all events $W_1 = \{e_1, \ldots, e_k\}$ such that $e_k.time - e_1.time \leq w$, whereas $e_{k+1}.time - e_1.time > w$. A second window, in turn, contains events $W_2 = \{e_m, \ldots, e_n\}$, such that $e_m.time - e_1.time \geq s$ and $e_{m-1}.time - e_1.time < s$, while also $e_m.time - e_n.time \leq w$ and $e_{m+1}.time - e_n.time > w$. A sliding, time-based window is illustrated in Fig. 3.

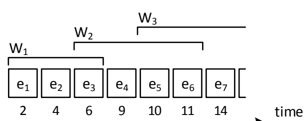

Fig. 3. Windows of size 5 and slide 4 over an event stream.

Applying the query operators (e.g., selections, aggregations) to each window yields a sequence of sets of result events. Those may be ordered again to generate an output event stream.

This query model is illustrated in Fig. 4. This query is applied to a stream of smart meter readings (`SMReadingStream`) with a `2h` window and a `1min` slide (assuming second granularity). For each window, the power values after the start of DR event (`DRStartTime`) are aggregated (`Sum`) and added to the last measurement (`Last`) to create an event in the output stream.

III. Stream Monitoring for Dynamic DR

This section first gives an overview of our approach to dynamic DR based on event stream monitoring (Section III-A). It then clarifies how the participants' compliance during a DR event is assessed (Section III-B), before turning to distributed, adaptive evaluation of the underlying monitoring queries (Section III-C).

```
Select Last(*) As smartMeterEvent,
       houseID, Sum(power) As houseAccPower
From SMReadingStream#ext_timed((time%60)==0, 7200)
Where time > DRStartTime
Group By houseID;
```

Fig. 4. Example monitoring query for a smart meter.

A. Overview

To realize dynamic DR that is based on a continuous assessment of the residents' energy consumption, this paper relies on event stream monitoring as illustrated in Fig. 5. That is, following the topology of a network, all DR participants are divided into groups to enable decentralized prediction of their compliance while a DR event unfolds. The compliance predictions on the group level are then used for compliance prediction at the global level by the utility. Based thereon, decisions on additional DR events are taken. In case of lossy connections within or across groups, fault tolerance mechanisms are employed to guarantee successful events' transmissions [8].

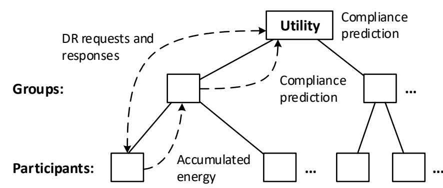

Fig. 5. Infrastructure for dynamic DR based on event stream monitoring.

Reflecting on the requirements for dynamic DR as outlined in Section I, the group-based approach to monitoring means that the approach is inherently distributed. As detailed below, the accumulation of energy measurements and the compliance prediction is further grounded in queries over event streams to meet the requirement of traceable and online processing.

B. Monitoring Queries used for Compliance Prediction

Using the above general setup, compliance to a DR task, locally by participants, but also globally on the level of the utility, is assessed as follows. A participant k is predicted at a time t_{test} to successfully comply with a DR task if

$$\sum_{t=t_{start}}^{t_{test}} (B_{k,t} - P_{k,t}) \geq \lambda_1, \tag{5}$$

where λ_1 is a threshold energy level, whose selection is case-specific, and depends on the total energy reduction requested. This simple compliance test could be made as sophisticated as the utility requires; for instance, it could use as additional inputs the accept/reject responses of the individual consumers. Further, the result of this method could be a probability instead of a binary comply/non-comply output. Finally, the aggregate system is said to be compliant if $\gamma_1\%$ of the total participants are compliant according to (5). With the above infrastructure, compliance may also be assessed at intermediate levels. That is, a group G of participants is predicted at a time t_{test} to successfully comply with the given DR task if

$$\sum_{k \in G} \sum_{t=t_{start}}^{t_{test}} (B_{k,t} - P_{k,t}) \geq \lambda_2. \tag{6}$$

Again, the overall system is compliant to the DR event if $\gamma_2\%$ of the groups are compliant according to (6). Note that γ_1 and γ_2 are system-dependent and chosen by the utility during the planning phase of the DR program.

The above model is implemented by means of event stream processing as follows. For each participant, the aforementioned monitoring query, Fig. 4, is evaluated. It accumulates the power measurements emitted by a resident's smart meter after the start of a DR event per sliding window. The output stream is referred to as `AccSMReadingStream`.

```
Select Count(Distinct houseID) < baselineNum As 'Alert'
From AccSMReadingStream
Where (baseline-houseAccPower)>lambda1
Group By time;
```

Fig. 6. Query to assess compliance based on individual participants.

Based thereon, the computation of (5) along with the global assessment of the number of compliant participants is realized by the query in Fig. 6. For each resident at each time point, it compares the accumulated power `houseAccPower` with the respective accumulated baseline `baseline`, and counts the number of compliant residents. If less than `baselineNum` (derived from γ_1) residents reach compliance, the system is non-compliant, so that an alert shall be emitted.

```
Select (groupBaseline-Sum(houseAccPower)) > lambda2 As '
    GroupAlert'
From AccSMReadingStream
Group By time;
```

Fig. 7. Query to assess compliance per group.

Exploiting the distribution of monitoring at the level of groups of participants, however, the query shown in Fig. 7 would be used for monitoring. It accumulates power per group and compares it against the respective baseline, realizing (6). The number of non-compliant groups is then determined in the same way as discussed above for individual participants.

C. Distribution and Adaptivity

To achieve scalability of the outlined monitoring solution to large-scale networks, the distribution induced by structuring the monitoring around groups of participants is of utmost importance. In a centralized setting, each resident's smart meter sends the accumulated energy to the utility at any point in time, which yields a global communication cost that is denoted by C_G. The utility would then compute the compliance prediction, which induces a computation cost $\Omega(n)$, with n as the number of measurement events. By exploiting distributed monitoring, both communication and computation costs are reduced.

Assume that N residents are divided into g groups, with the i-th group having $|G_i|$ residents. The local data communication cost within a group is denoted as C_L, where $C_L < C_G$.

In centralized monitoring, at a single point in time, the system-wide one-way communication cost is $N \times C_G$ and the computation cost is $\Omega(N)$. This yields a monitoring cost of $C_c = N \times C_G + \Omega(N)$ per time instant.

In a distributed case, however, groups perform local communication and the computation of compliance prediction in parallel. At any time instance, the system-wide monitoring cost is $C_d = \sum |G_i| \times C_L + \Omega(\max(|G_i|))$, where G_i is a non-compliant group. This monitoring can be adapted while a DR event unfolds. For a group that is compliant, local communication and computation of local compliance predictions are neglected. The idea here is that within a group, the highly committed residents' reduction may offset their non-committed counterparts energy, making the whole group reach the required compliance level. In addition, once the whole system is found to be compliant, the measurement rate at each of the residents' smart meters is reduced, as the data is now solely used for billing purposes. In practice, this is realized by simply increasing the slide of the window of the query evaluated directly at the smart meter, see Fig. 4. As shown in the remainder, such adaptations lead to significant reductions of the overall monitoring cost.

IV. Performance Evaluation

This section presents experimental results to highlight the effectiveness of dynamic DR as well as the efficiency of its realization when relying on event stream monitoring.

A. Experimental Setup

Dynamic DR is implemented for a residential distribution system whose topology is based on the standard IEEE 13 node test feeder [9]. The number of residences at each of the nodes with spot-loads is calculated by assuming that each home contributes 5kW on an average to the rating. In this way, 654 residences in total are assumed to be distributed over the various nodes of this system. As explained in Section II-A, baseline estimation is essential in determining the DR response of each participant. However, since this paper relies on simulated load profiles with and without DR, for ease, the baseline demands are directly obtained using (2) and (3) with $degComp = 0$.

B. Effectiveness of Dynamic DR

Assume that the utility wishes to reduce the system peak demand during the period 7-9PM, by providing an incentive $u = 0.5$ to all participants. Further, assume that all consumers accept the given DR request, and the individual $degComp$ values are uniformly distributed in $[0.7, 0.9]$. Implementing the above simulation procedure in MATLAB, the benefit of dynamic DR is now illustrated. Say the utility predicts 30 minutes into the DR event that the response would be unsatisfactory for some previously-determined values of γ_1 and γ_2. It then generates a second DR request providing an increased incentive ($u = 0.7$), which is communicated to half of the total population (chosen at random).

Fig. 8 shows the system-wide demands for three cases: (i) the baseline case with no DR; (ii) the traditional open-loop implementation of a single DR event; and (iii) feedback

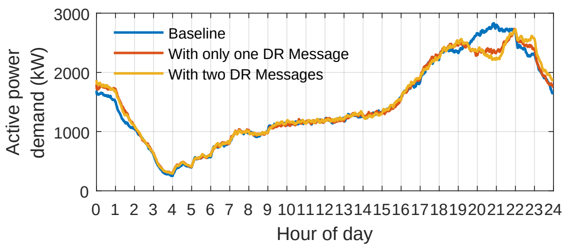

Fig. 8. Impact of feedback on the system-wide demand for a DR event between 7-9PM (single run of simulation). $degComp$ for the population is distributed uniformly in $[0.7, 0.9]$. Incentive u offered by the utility is 0.5 and 0.7 respectively for the first and second DR messages.

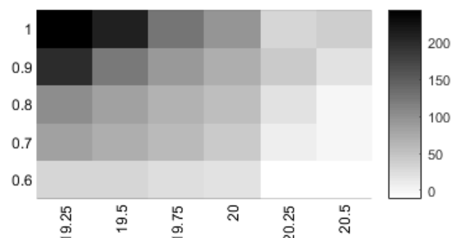

Fig. 9. Heatmap (averaged over 100 simulation runs) depicting the increased system-wide energy reduction (kWh) due to the second DR event for varying times of feedback (in X-axis, hours) and incentive u (in Y-axis).

implemented, with a second DR request sent to 50% of the population after 30 minutes. The values of $degComp$ for the residents are assumed not to vary on the receipt of the second message; only their reaction to the incentive u varies. This assumption is sufficient for assessing the effectiveness of dynamic DR; the exact variation in the consumer behavior due to this incentive could be modeled as detailed as the case may require, depending on the propensity of each population set to being swayed by the additional monetary benefit offered.

The system-wide energy reduction (averaged over 100 simulation runs) during the DR event when compared to the baseline demand is 426.11kWh, when only the first DR request is sent. However, it increases to 498.27kWh with the second DR event. Variation in the increased energy reduction for different times of sending the second request, and the incentive offered, is shown in Fig. 9. The sooner the second request is delivered, the higher are the energy savings. Further, a smaller incentive offered early on into the first DR event could provide the equivalent response as a higher incentive offered later on, thereby underscoring the need and benefit of early prediction of the success of the response to a given DR request.

C. Communication and Computation Efficiency of Event Stream Monitoring

To test the efficiency of the proposed realization of dynamic DR using event stream monitoring, a prototype is implemented based on the Esper [10] engine. Using the above simulation data, this study considered a topology that divides the 654 residents into 21 groups. Per group, a baseline with minute granularity was precomputed from simulation data. The queries of Fig. 4 and Fig. 7 are evaluated per resident and group, respectively.

Centralized monitoring is used as a baseline for comparison; here, measurements are sent by each smart meter to the utility, which evaluates the query in Fig. 6 for compliance prediction.

TABLE I
MONITORING COST COMPARISON, CENTRALIZED VS. DISTRIBUTED

	7-9PM		8-9PM	
	Centralized	Distributed	Centralized	Distributed
Communication	78,480 $\times C_G$	39,953 $\times C_L$	39,240 $\times C_G$	713 $\times C_L$
Computation	78,480 $\times C_E$	1,553 $\times C_E$	39,240 $\times C_E$	591 $\times C_E$

C_G and C_L are the abstract units for global communication cost and local communication per event transmission; C_E is the abstract unit for computing predication per event.

Now consider event stream monitoring for the time between 7–9PM with compliance predication taking effect from 8PM. When a group reaches local compliance, monitoring is adapted so its data is no longer forwarded to the utility. As part of our experiments, it was observed that around half of the groups are compliant in the time period between 8–9PM. Communication cost is measured by the data volume transmitted in the network, and computation cost by the number of events used for compliance prediction, denoted as C_E. As explained in Section III-C, groups compute their predictions in parallel, so that the maximum cost over all groups at a time instant is considered when aggregating the costs for a whole DR event.

A comparison of the costs for centralized and distributed monitoring for the periods 7-9PM and 8-9PM is given in Table I. Computation cost is reduced by 50$\times$ and 66$\times$ for the two periods respectively. Assuming $C_G = 5 \times C_L$, reductions of communication costs are then 9.8$\times$ and 275$\times$ respectively. Clearly, adaptive distributed event stream processing reduces monitoring costs by at least one order of magnitude.

V. CONCLUSION

This paper illustrated the merits of feedback in an incentive-based DR implementation. This feedback loop, from the utility to the participants, has a dynamic incentive structure at its core, and is employed during the course of a DR event to ensure compliance of the system to the required level demanded by the utility. A scalable realization of such a loop has been presented using distributed, adaptive event stream monitoring.

ACKNOWLEDGMENT

This work was partially funded by the NUS-HU Profile Partnership programme.

REFERENCES

[1] G. Schuitema, L. Ryan, and C. Aravena, "The consumer's role in flexible energy systems: An interdisciplinary approach to changing consumers' behavior," *IEEE Power Energy Mag.*, vol. 15, no. 1, pp. 53–60, 2017.
[2] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, p. 378, 2000.
[3] M. Jain, *et. al*, "Methodologies for effective demand response messaging," in *IEEE Int. Conf. Smart Grid Commun.*, 2015, pp. 453–458.
[4] M. L. Goldberg and G. K. Agnew, "Measurement and verification for demand response," *US DOE: Washington, DC, USA*, 2013.
[5] KEMA Incorporated, "2009 California residential appliance saturation study," 2009.
[6] M. Dayarathna and S. Perera, "Recent advancements in event processing" in *ACM Comput. Surv.*, 2018, pp. 33:1–33:36.
[7] A. Arasu, *et. al*, "STREAM: the Stanford data stream management system," in *Data Stream Management*, Springer, 2016, pp. 317–336.
[8] A. Martin, *et. al*, "User-Constraint and Self-Adaptive Fault Tolerance for Event Stream Processing Systems," in *IEEE/IFIP Int. Conf. Depend. Syst. Networks*, 2015, pp. 317–336.
[9] W. H. Kersting, "Radial distribution test feeders," in *IEEE PES Winter Meeting. Conf. Proc. (Cat. No.01CH37194)*, vol. 2, 2001, pp. 908–912.
[10] Esper. [Online]. Available: http://www.espertech.com/esper/