Cloud-based Demand Response for Smart Grid: Architecture and Distributed Algorithms

Hongseok Kim, Young-Jin Kim, Kai Yang, and Marina Thottan
Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA
{hongseok.kim, young.jin_kim, kai.yang, marina.thottan}@alcatel-lucent.com

***Abstract*—In this paper we propose cloud-based demand response (CDR), a novel demand response architecture for fast response times in large scale deployments. The proposed architecture is in contrast to master/slave based demand response where the participants directly interact with the utility using host address-centric communication. CDR leverages data-centric communication, publisher/subscriber and topic-based group communication to make demand response secure, scalable and reliable. To the utility, CDR appears to be a black box function call that takes an input from the utility, e.g., power deficit and gives an output to the utility, e.g., power reduction per customer and the corresponding price incentive. Using this implementation framework, we propose two market-based distributed algorithms (bisection and Illinois methods). The proposed algorithms exhibit at least exponentially fast convergence with $O(1)$ iteration as the number of customers grows and outperform prior work of the dual gradient method in terms of convergence speed while keeping the same messaging overhead.**

I. Introduction

World-wide research initiatives have just started to transform the old and unintelligent power grid into a new grid, called *smart grid*. As part of this transformation the underlying information and communication technology is expected to contribute significantly to improving the quality, reliability, security and efficiency of the power grid. The critical function of an electric power grid is to balance the supply and demand of electricity at any instance; either if the demand exceeds supply, or the supply exceeds demand, both these situations seriously threaten the stability of the grid, and therefore power generation must follow load accurately.

Peak power usage during hot summer (or cold winter) days is one of the biggest concerns in electric power system, and to meet the peak demand, high marginal costs are incurred to maintain stand-by power sources. However, in addition to the high investment and running cost, stand-by generators are mostly based on fossil fuel, and thus increases the carbon footprint. Hence, rather than increasing the *physical* power generation facilities, (which runs only for a limited amount of time per year), a mechanism that enables *virtual* power generation is being considered. This technology is called *demand response* where electricity customers actively participate in balancing the supply and demand curve [1]. Under demand response, end-use customers may voluntarily reduce their electric power consumption based on real-time price or incentive price signals. The virtually generated electricity from demand response is sometimes called *negawatt* in the sense that the reduction of load is equivalent to power generation of the same amount.

There are several mechanisms to realize demand response. The simplest one is direct load control where the utility controls the customer's load based on advanced contracts; in this case when the power imbalance arises, the utility turns off or controls some types of loads: water heater, thermostat, pool pump, etc. Another mechanism of demand response is based on time-dependent price where electricity price changes over time so that customers can adapt their usage to minimize their total electricity bills. This is an indirect method, and there always exists uncertainty about the amount of achievable load reduction. Incentive (or bidding) based demand response provides more deterministic load reduction by exploiting bidding and bargaining process, i.e., market-based mechanism; customers submit their bids (the amount of load reduction and desired incentive price), or alternatively, the utility offers an incentive price ($/kW) to encourage customers to commit their load reductions. In this paper, we mostly focus on the incentive-based demand response. As observed in [4]–[6], the bidding process/algorithm for incentive-based demand response is a distributed optimization process. We note that, however, a proper demand response architecture is crucial in addition to the optimization algorithm. The requirements for the demand response architecture are as follows.

- *Security*: Since demand response requires interactions between the utility and customers, the message exchange between server and clients must be secure. Otherwise, demand response is vulnerable to cyber attack, which threatens grid stability.
- *Reliability*: Demand response should be free from a single point of failure (e.g., the server breakdown) or bottlenecks in running algorithms.
- *Scalability*: Demand response should be scalable so that a large number of customers, including residential customers, can participate in the program.
- *Speed*: Matching supply and demand is more challenging than ever, as more renewable energy sources such as solar photo voltaic and wind farms with variable power output are introduced to smart grid. Fast demand response is becoming an essential ancillary service for the power grid.
- *Efficiency*: If demand response is based on market mechanism, it is desirable to achieve the objectives of all participants; utility minimizes the cost of realizing de-

TABLE I
COMPARISON OF DEMAND RESPONSE ARCHITECTURES

	Master/Slave-based Demand Response	Cloud-based Demand Response
Communication	Host address-centric communication	Data-centric communication [2], [3]
Network Architecture	Master and slave (*static tree*-based)	Publisher and subscriber (*cloud*-based)
Reliability	Vulnerable to a single point (server) of failure	Hard to attack subscribers
Traffic concentration	Server side	Dispersed through the cloud
Scalability	Number of nodes is limited by server capacity	Highly scalable
Computation	Between master (utility) and slaves (customers)	Within customers and/or cloud
Latency	Determined by RTT between master and slaves	Low latency when nodes are concentrated
Utility's role	Utility manages demand response	Utility sees demand response as a black box
Drawback	Initially simple, but as the number of nodes grows scalability becomes an issue	Having overhead for a small size network, but scalable

mand response, customers maximize their profits, and the regulator maximizes the social welfare.

This paper addresses both architectural and algorithmic aspects for a large scale and fast demand response. We propose a new architecture called *cloud-based demand response (CDR)*. CDR leverages our ongoing work on the distributed smart grid information infrastructure designed for accommodating a broad range of smart grid applications [7]; among them, this paper specifically focuses on the demand response application.

We first describe how CDR satisfies the above mentioned requirements of security, scalability and reliability from an architectural perspective. We then address optimization algorithms for scalability, speed and efficiency. We note that the dual gradient method [4], [6] may not exhibit fast convergence; since the convergence speed is very dependent on choosing the step size, which is not always possible, specifically when the cost functions of customers are kept confidential. Alternatively, we propose the bisection method and Illinois method that significantly improve the convergence speed with the same message exchange overhead, without the need to know the cost functions.

II. DEMAND RESPONSE ARCHITECTURE

We first overview the traditional demand response architecture and motivate the design of cloud-based demand response leveraging a distributed smart grid information infrastructure.

A. Master/slave-based demand response

To the best of our knowledge, most demand response systems proposed so far have been based on master/slave architecture. Utility's energy management system (EMS) interacts with customers's EMS individually. Basically, master/slave architecture is host-address centric communication and is good for a small scale network due to its simplicity. However, from system protection perspective, master/slave architecture, for demand response, has several potential drawbacks [7]. When the communication occurs on IP-enabled networks, it is possible that meters and home EMSes can be compromised by cyber attackers because these systems are usually installed outside the physical security perimeter of the utility. In master/slave based demand response systems, all slave nodes (e.g., meters, home EMS) must know the IP addresses of the designated master nodes in utility sides. Therefore from a security perspective, applying the traditional approach for demand response causes extensive exposure to cyber-attacks such as distributed denial of services (DDoS) from compromised slave nodes.

From reliability perspective, it is well known that a single point of failure is one of the biggest concerns in master/slave architecture; for example, DDoS is a widely used method to collapse a server. From scalability perspective, the maximum number of clients are limited by the server's capacity. Furthermore, when demand response operates as an iterative process, the communication latency between a master and slaves can be high. The latency gets even longer when a server is protected by firewalls or intrusion detection systems. If the utility wants to deploy a large-scale demand response program, the utility's EMS server must be able to resolve the potential problems listed so far. Table I summarizes the features of master/slave based demand response.

B. Cloud-based demand response

Motivated by above-mentioned considerations, we propose a new concept of CDR.

Data-centric communication: The underlying technology of CDR is the information infrastructure proposed in [7] leveraging *data-centric* communication [2]. Data-centric communication is conceptually the opposite of *host address-centric* communication where the sender(s) and the receiver(s) need to know their addresses (e.g., IP address) for communication. Data-centric communication does not require the explicit knowledge of each other's address but is more associated with the data. Message exchanges without relying on IP address visibility makes CDR robust to cyber attacks.

Definition 1 (Cloud): A *cloud* is a collection of computing entities (or *hosts* hereafter) with memory and/or storage. The cloud is essentially an *overlay network*, which constitutes a graph $G(V, E)$ having the hosts $v \in V$ as vertices and the transport layer connection $l \in E$ between two neighboring hosts as edges.

The utility and customers interact through the cloud, and the functions for realizing demand response are performed in a cloud rather than in the utility's EMS. Hence, the role of utility is minimized. From utility's perspective, CDR appears to be a *black box information system* that takes an input from utility (e.g., the amount of power deficit) and gives an output to utility (e.g., how much to reduce loads per customers and at which incentive price). Hence, the utility need not be concerned about issues regarding security, scalability and reliability.

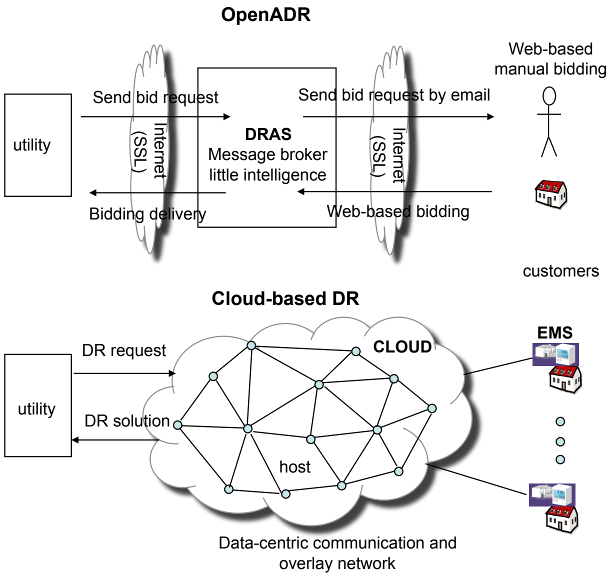

Fig. 1. OpenADR vs Cloud-based DR.

Topic-based group communication: The sender(s) and the receiver(s) communicate with each other without knowing the IP address. This becomes possible by leveraging the concept of publisher/subscriber and *topic-based group communication* [8]. Here by topic we mean the type of data to be delivered. For example, in CDR, the incentive price broadcasted to the customers can be one topic. The bidding information from the customers to the cloud can be another topic. In the publisher/subscriber model, publishers announce the availability of one topic, and subscribers announce their interest in it. The matching of the publishers and subscribers for delivering data regarding a particular topic is made by a distributed hash function, which is shared by all cloud participants. The role of hash function is to designate on the overlay network the rendezvous host(s) on which publishers write data, and from which subscribers read the data. Unlike the case of master/slave-based demand response where data traffic is concentrated at the server, CDR disperses data traffic throughout the cloud by properly designing the hash function.

The routing algorithm in the overlay network and the design of hash function for load balancing in the cloud, are beyond the scope of this paper. We assume that there is a distributed routing algorithm and an optimally designed hashing system in the overlay network [9] and focus only on the optimization algorithms.

C. OpenADR vs. CDR

Fig. 1 shows the Open Automated Demand Response (OpenADR) [10] and CDR architecture. OpenADR is the communication specification describing an open standard-based communication data model for information exchange between demand response participants. The utility and the customers access the demand response access server (DRAS) over the Internet, which transparently delivers messages. The utility sends the bidding request, which is delivered to the customers by email, and the customers place their bids on the DRAS web server, which is in turn delivered to the utility. This process requires the interaction of humans at the customer side. OpenADR is based on IP communication and has the traditional master/slave architecture. As can be seen in Fig. 1, CDR looks similar to OpenADR; DRAS is replaced by the cloud of hosts, which implies that CDR can in principle support OpenADR. However, recall the features of the data-centric communication, IP address invisibility, topic-based group communication, and publisher/subscriber communication model, all of which are considered to realize secure, reliable and scalable smart grid information infrastructure for demand response application

III. System Model

A. Assumption

We consider a large scale demand response system where the number of customers is huge so that even residential customers consuming small amount of power participate in the program. Suppose that utility (e.g., Con Edison) covers some geographical area (e.g., New York state) and keeps track of power consumption of customers. On a hot summer day, for example, the utility observes that load is rapidly increasing and more power supply (or less power demand) is required. Utility then has two options: 1) purchase more power from spot market by paying the *spot market price*. Note that spot market price is highly volatile and changes rapidly, sometimes more frequently than an hourly scale; 2) invoke demand response program if the incentive price could be quoted to be less than the spot market price. Here, utility wants to determine the right incentive price that achieves the desired load reduction.

We assume that the utility divides the demand area into regions $j \in \mathcal{J}$ based on power grid topology to apply different demand response (i.e., different incentive prices) to different regions, e.g., per substation or feeder. This is because, in power grid, when power shortage and/or outage happens, it is crucial to prevent cascading failure by isolating the region where the power problem occurred. Suppose that the utility can estimate accurately the power deficit region by region, denoted by $\{D_j | j \in \mathcal{J}\}$. Utility then wants to know a set of optimum incentive prices $\{\lambda_j | j \in \mathcal{J}\}$. Let $i \in \mathcal{N}_j$ be a customer in region j.

Assumption 3.1 (price-taker): Since we address a large number of customers, $N_j = |\mathcal{N}_j|$, $\forall j \in \mathcal{J}$ is assumed to be large enough, and the expected load reduction of each customer denoted by x_i is negligible compared to D_j. Hence, each customer behaves as a *price-taker* in determining the load reduction x_i and cannot exercise market power.

Given the incentive price λ_j, customer i decides the optimal x_i that maximizes its profit by solving the following optimization problem,

$$\begin{array}{lll} \textbf{Customer } i \in \mathcal{N}_j\textbf{:} & \text{maximize} & \lambda_j x_i - c_i(x_i) \\ & \text{variable} & x_i \end{array} \quad (1)$$

where $c_i(x_i)$ is a *cost* function or disutility function experienced by customer i in reducing its power consumption by x_i. In this paper, we assume that $c_i(x_i)$ is monotone increasing, strictly convex, differentiable in x_i, and $c_i(0) = 0$. Then, if

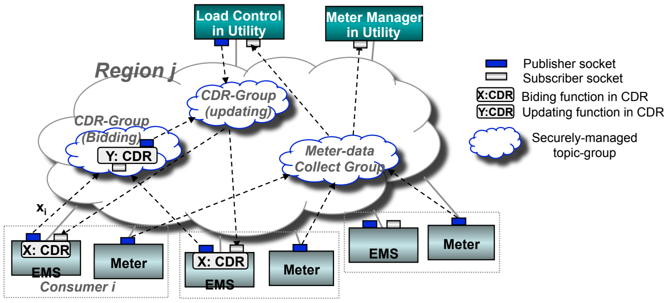

Fig. 2. Topic-based group communication in the cloud.

$c_i'(x_i)$ is invertible, the optimal load reduction x_i is given by

$$x_i = c_i'^{-1}(\lambda_j)$$

from the first order condition. Note that the cost function is time-varying because the discomfort experienced by customers who cannot run, for example, air conditioner depends on the outside temperature. We assume that all customers participating in demand response have their own EMS and know $c_i(x_i)$ based on historical usage pattern in response to the real-time price signal. It should be noted that inferring $c_i(x_i)$ is a critical research topic in realizing *autonomous* demand response.

B. Demand response optimization

The demand response optimization problem can be posed as:

$$\begin{aligned} \textbf{Utility:}\quad & \text{minimize} && \sum_{j\in\mathcal{J}} \lambda_j D_j && (2)\\ & \text{subject to} && \sum_{i\in\mathcal{N}_j} x_i = D_j, \forall j \in \mathcal{J} && (3)\\ & && x_i = x_i(\lambda_j) && (4)\\ & \text{variables} && \{\lambda_j \le \Lambda_j | j \in \mathcal{J}\}. && \end{aligned}$$

The optimization variables are the incentive prices $\{\lambda_j \le \Lambda_j | j \in \mathcal{J}\}$ where Λ_j represents the price of the alternative solution to maintain grid stability, e.g., the (forecasted) spot market pice. Note that x_i in (4) is given by solving (1).

We see that problem **Utility** can be decomposed into $|\mathcal{J}|$ subproblems and can be solved separately for each region because the constraints are not coupled. Furthermore, one can show that minimizing $\lambda_j D_j$ for region $j \in \mathcal{J}$ is equivalent to solve the following problem using $\lambda_j \le \Lambda_j$ as a Lagrange multiplier,

$$\begin{aligned} & \text{minimize} && \sum_{i\in\mathcal{N}_j} c_i(x_i) && (5)\\ & \text{subject to} && \sum_{i\in\mathcal{N}_j} x_i = D_j && (6)\\ & \text{variables} && \{x_i | i \in \mathcal{N}_j\}. && (7) \end{aligned}$$

IV. Distributed Operation of CDR

A. Demand response bidding diagram

Fig. 2 shows the cloud-based operation. In a region $j \in \mathcal{J}$, a meter-data collection group is responsible for collecting power consumption data. As can be seen, meters periodically publish the measured power consumption to the meter-data collection group. The load controller in the utility is subscribed to this information. Based on measurements, once the demand is expected to exceed power system capacity, demand response is automatically invoked in that region by the utility publishing (D_j, Λ_j) to region $j \in \mathcal{J}$. One might wonder that, if the utility reveals their maximum affordable price Λ_j to the customers, that would make the customers deviate from solving (1) and strategically respond with some other quantities associated with fake cost functions. However, price-taking behavior indeed maximizes their profits, see Assumption 3.1, if the customers cannot exercise market power.

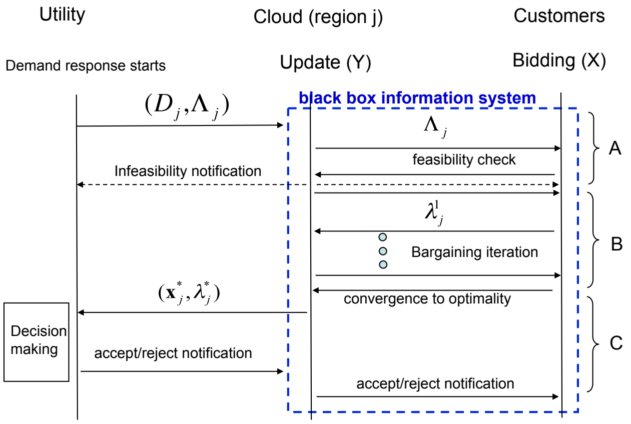

Fig. 3. Cloud-based Demand Response Bidding Diagram

Once CDR is invoked, it proceeds three steps as can be seen in Fig. 3. Step A is the feasibility check. The cloud publishes the initial incentive price $\lambda_j^0 = \Lambda_j$ to the customers. If $\sum_i x_i(\lambda_j^0) < D_j$, the cloud notifies the infeasibility to the utility, and the demand response terminates. Otherwise, it goes to the step B, the iterative bargaining process. Then, a set of customers in region j *autonomously* compute how to collectively reduce load by D_j and report to the utility a pair of information $(\mathbf{x}_j^*, \lambda_j^*)$ where $\mathbf{x}_j^*$ is an optimal vector $\{x_i^* | i \in \mathcal{N}_j\}$ denoting the individual commitment of load reduction and λ_j^* is the *suggested* incentive price (\$/kW). In step C, the utility finally has two options: either take the offer from the customers or seek an alternative (e.g., spot market) solution based on λ_j^*. Even though this paper does not address the regulatory issue, that can be considered; for example, the regulatory may recommend the use of demand response from *greening* perspective and thus subsidizes so that the utility may set Λ_j higher than the spot market price.

B. Topic-based group communication

The bargaining iteration in step B is indeed the distributed optimization process; the cloud and the customers iteratively negotiate the incentive price based on two functions: the bidding function X embedded in the customer's EMS and the price update function Y embedded in a host within the cloud. As can be seen in Fig. 2, while the iteration goes on, at k-th iteration, the function X gets λ_j^k from a topic group *CDR Update Group*. After computing x_i^k from (1), the function X injects x_i^k into a topic-group *CDR-Bidding Group*. At every iteration, a host is selected by a hash function, performs the Y function, and publishes the next incentive price λ_j^{k+1} to all participating customer EMSes. The choice of the host may change per iteration depending on the hash function and considering the tradeoff between security and host randomization overhead.

C. Price update function Y in a host within the cloud

We next discuss the implementation of Y. In fact, CDR can be implemented by leveraging various distributed optimization algorithms; for example, the dual gradient method [4], [6]. However, the dual gradient method could exhibit slow convergence since the convergence depends on the choosing the right step size; this could be difficult when the customers keep their cost function confidential. Newton method exhibits quadratic convergence speed but it requires the second derivative operation of the cost function (i.e., other than function X) as well as twice of feedbacks per iteration than the dual gradient [5]. To keep the functionality at the home EMS simple, we consider the distributed bisection and the Illinois method, showing faster convergence than the dual gradient with the same amount of feedback per iteration.

1) Bisection method: The proposed bisection algorithm neither requires any knowledge of the cost functions nor the determination of the right step size, but guarantees exponentially fast convergence. The algorithm starts with $[\lambda_{j,\min}$, $\lambda_{j,\max}]$, which defines the searching range of the optimal incentive price. Initially, $\lambda_{j,\min} = 0$, and $\lambda_{j,\max} = \Lambda_j$. The iteration goes as follows. At k-th iteration, the incentive price is published as $\lambda_j^k = \frac{\lambda_{j,\max}+\lambda_{j,\min}}{2}$, and x_i^k are in turn injected into the cloud . Then, $[\lambda_{j,\min}$, $\lambda_{j,\max}]$ is updated as follows. If $\sum_{i\in\mathcal{N}_j} x_i^k > D_j$, the incentive price is higher than the optimal price so the committed load reduction is more than what is needed. Hence, the incentive price should be reduced, and $\lambda_{j,\max} = \lambda_j^k$. Similarly, if $\sum_{i\in\mathcal{N}_j} x_i^k < D_j$, the incentive price is lower than the optimal price so the committed load reduction is less than what is needed. Hence, the incentive price should be increased, and $\lambda_{j,\min} = \lambda_j^k$. This iteration goes until convergence.

2) Illinois method: In addition to the bisection method, we could also use the Illinois method for faster convergence. Similar to the bisection method, Illinois method starts with an initial bracket $[\lambda_{j,\min}, \lambda_{j,\max}]$ and progressively reduces the search space. Initially we set $\lambda_{j,\min} = 0$ and $\lambda_{j,\max} = \Lambda_j$. For a given incentive price value λ_j, we aim to calculate a set of optimal load reduction values to minimize a Lagrangian function, i.e.,

$$g(\lambda_j) = \inf_{\mathbf{x}_j} L(\mathbf{x}_j, \lambda_j), \tag{8}$$

where $L(\mathbf{x}_j, \lambda_j)$ is given by

$$L(\mathbf{x}_j, \lambda_j) = \sum_{i\in\mathcal{N}_j} c_i(x_i) + \lambda_j(D_j - \sum_{i\in\mathcal{N}_j} x_i). \tag{9}$$

Recall that x_i^k corresponds to the load reduction that the customer i is willing to commit after receiving the incentive price λ_j^k from the utility at the k^{th} iteration. Since $c_i'(x_i^k) = \lambda_j^k$, we have $x_i^k = x_i(\lambda_j^k)$. Notice that the existence of the inverse function follows from the assumption that the cost function is strictly convex. Hence, $g(\lambda_j)$ can be written explicitly as follow,

$$g(\lambda_j^k) = \sum_{i\in\mathcal{N}_j} g_i(\lambda_j^k) + \lambda_j^k D_j, \tag{10}$$

where

$$g_i(\lambda_j^k) \;=\; c_i\left(x_i(\lambda_j^k)\right) - \lambda_j^k x_i(\lambda_j^k), \tag{11}$$

and its derivative is given by

$$\begin{aligned} g_i'(\lambda_j^k) &= c_i'\left(x_i(\lambda_j^k)\right) x_i'(\lambda_j^k) - x_i(\lambda_j^k) - \lambda_j^k x_i'(\lambda_j^k) \\ &= -x_i(\lambda_j^k). \end{aligned} \tag{12}$$

Furthermore, the derivative of $g(\lambda_j)$ can be compactly represented as

$$g'(\lambda_j^k) = D_j - \sum_{i\in\mathcal{N}_j} x_j^k. \tag{13}$$

Under the assumption that the cost function is strictly convex, it turns out the derivative of $g_i(\lambda_j^k)$ is just the negative of the committed load reduction, i.e., $-x_i^k$. This implies that at each iteration the Illinois method requires the same X function as the bisection and the dual gradient algorithms. Upon collecting all x_i^k, the next incentive price λ_j^{k+1} is calculated as follows,

$$\lambda_j^{k+1} = \frac{\lambda_{j,\min} g'(\lambda_{j,\max}) - \lambda_{j,\max} g'(\lambda_{j,\min})}{g'(\lambda_{j,\max}) - g'(\lambda_{j,\min})}. \tag{14}$$

This is based on the root-finding algorithm of $g'(\lambda_j) = 0$. The new incentive price is then published to all customers. After that, the cloud collects the committed load reduction x_j^{k+1} from all customers and compute $g'(\lambda_j^{k+1})$ according to (13). If $g'(\lambda_j^{k+1}) < 0$, then $\lambda_{j,\min}$ is retained, and $\lambda_{j,\max} = \lambda_j^{k+1}$. If $g'(\lambda_j^{k+1}) > 0$, then $\lambda_{j,\min} = \lambda_j^{k+1}$, and $\lambda_{j,\max}$ is retained. This process repeats until convergence. However, it is known that such an approach may lead to slow convergence, i.e., one end-point is retained permanently. A better alternative of (14) is as follows; if $\lambda_{j,\max}$ is retained twice or more in a row, $\frac{1}{2}g'(\lambda_{j,\max})$ is used instead of $g'(\lambda_{j,\max})$ in (14). Similarly, if $\lambda_{j,\min}$ is retained twice or more in a row, $\frac{1}{2}g'(\lambda_{j,\min})$ is used instead of $g'(\lambda_{j,\min})$ in (14). It was shown than $\frac{1}{2}$ scaling helps preventing either of the end points from being retained permanently. The convergence property of the Illinois method is summarized in the following lemma [11].

Lemma 1: The Illinois method converges exponentially fast to the optimal solution if the optimal solution falls into the initial bracket $[\lambda_{j,\min},\ \lambda_{j,\max}]$. In addition, the order of convergence is superlinear.

As the order of convergence of the bisection method is 1, the Illinois method converges faster than the bisection algorithm. Also, during each iteration of the Illinois method, upon receiving the incentive price information from the cloud, each customer only needs to feedback a single message, i.e., the derivative of $g_i(\lambda_j)$ in (12). In addition to its superlinear convergence rate, given $g'(0) = D_j$ and $g'(\Lambda_j)$ is known at step A, the Illinois method can achieve the optimal solution within one iteration for a quadratic function.

Proposition 4.1: If $c_i(x_i)$ is quadratic, the Illinois method converges to the optimal solution within one iteration.
Due to the space limitation, the proof is omitted in this paper.

Algorithm	Subgradient Algorithm	Bisection Algorithm	Illinois Algorithm	Newton's Method
Rate of convergence	Sublinear	Linear	Superlinear	Quadratic
Messages per iteration	1	1	1	2

TABLE II
COMPARISON OF DEMAND RESPONSE ALGORITHMS.

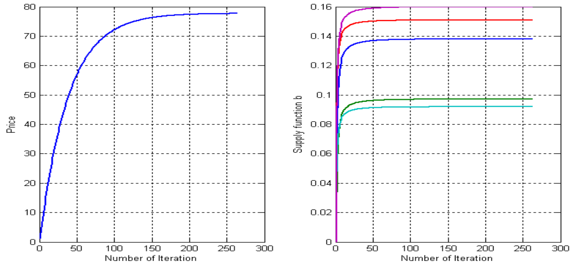

Fig. 4. Iteration using dual gradient.

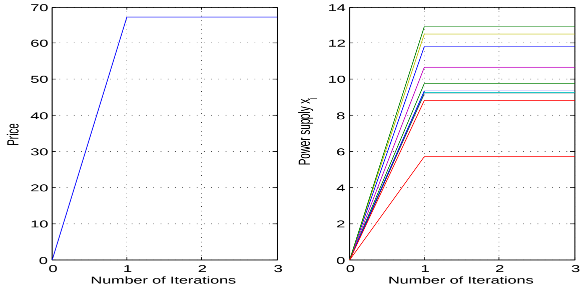

Fig. 5. Iteration using Illinois method.

D. Comparison of dual gradient, bisection and Illinois methods

To demonstrate the convergence properties of distributed algorithms, iteration results are provided here, specifically focused on the convergence speed. We use the same class of cost functions as in [4] where quadratic cost function is assumed: $D_j = 100$ and $c_i(x_i) = a_i x_i + b_i x_i^2$ where a_i and b_i are randomly chosen over $[1, 2]$ and $[2, 6]$ respectively. For dual gradient method, we use the same step size of 0.02 as in [4]. It should be noted that the dual gradient, bisection and Illinois methods are all scalable, i.e., the number of iterations does not grow as the number of customers increases; we experimentally verified up to $N_j = 1,000,000$. Due to the space limitation, however, we only provide the result with $N_j = 10$ as in [4] for the comparison of convergence speed between the dual gradient method of [4] and our proposed two methods. Fig. 4 shows the iteration for the dual gradient method. As can be seen the convergence requires more than 250 iterations; left figure shows the evolution of the incentive price, and the right figure shows the supply function bidding parameters of five randomly selected customers. Due to the space limitation we do not provide the simulation result for the bisection method; under the same set-up it exhibits much faster convergence than the dual gradient, i.e., about 10 iterations. Fig. 5 shows the convergence behavior of the Illinois method. As evident from the figure, the Illinois method achieves the best convergence performance and attains the optimal solution within one iteration. The performance of different demand response algorithms are summarized in Table II, with focus on the convergence behavior and messaging overhead per iteration. Here we use the amount of message passing required per customer per iteration to characterize the distributiveness of the algorithm [12]. As evident from the Table II, the global convergence property, the fast convergence speed, as well as the less amount of feedback make the Illinois method a possibly better alternative than other algorithms.

V. CONCLUSION

In this paper we proposed a new architecture called cloud-based demand response for fast and large-scale demand response. Unlike the demand response based on master/slave architecture, the proposed cloud architecture leverages data-centric communication, publisher/subscriber and topic-based group communications for secure, scalable and reliable purposes. The proposed system is attractive to the utility because the utility is spared from the specific details of implementation, which is performed as a cloud service. In realizing CDR, we solved the demand response optimization problem in a decentralized manner, and proposed the bisection and Illinois methods, which exhibit linear and superlinear convergence speed and outperform the dual gradient method with the same messaging overhead.

REFERENCES

[1] US Department of Energy, "Benefits of demand response in electricity markets and recommendations for achieving them," *Report to the United States Congress, available at http://eetd.lbl.gov*, Feb. 2006.

[2] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, "Networking named content," in *CoNEXT*, 2009.

[3] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker, "GHT: A geographic hash table for data-centric storage," Atlanta, Georgia, USA, Sept. 2002.

[4] L. Chen, S. H. Low, and J. C. Doyle, "Two market models for demand response in power networks," in *IEEE SmartGrid Comm.*, Oct. 2010.

[5] A. Papavasiliou, H. Hindi, and D. Greene, "Market-based control mechanisms for electric power demand response," in *Conference on Decision and Control*, Dec. 2010.

[6] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *Proc. IEEE SmartGridComm.*, 2010.

[7] Y.-J. Kim, M. Thottan, V. Kolesnikov, and W. Lee, "A secure decentralized data-centric information infrastructure for smart grid," *IEEE Comm. Mag.*, Nov. 2010.

[8] Rachid Guerraoui Patrick Eugster, Pascal Felber and Anne-Marie Kermarrec, "The Many Faces of Publish/Subscribe," vol. 35, no. 2, pp. 114–131, 2003.

[9] Young-Jin Kim, Marina Thottan, Hongseok Kim, and Gary Atkinson, "A geometrically inspired data-centric platform for secure, resilient smart grid communications," in *submitted to ACM SIGCOMM ICN*, 2011.

[10] Lawrence Berkely National Laboratory and Akuacom, "Open automated demand response communications specification," Apr. 2009.

[11] J.A. Ford, *Improved Algorithms of Illinois-type for the Numerical Solution of Nonlinear Equations*, Technical Report, CSM-257, University of Essex, 1995.

[12] K. Yang, Y. Wu, J. Huang, X. Wang, and S. Verdu, "Distributed robust optimization for communication networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, Sept. 2008.