Network Working Group	C. Jennings
Internet-Draft	Cisco
Intended status: Standards Track	B. Nordman
Expires: January 11, 2012	Lawrence Berkeley National Laboratory
	July 10, 2011

Communication of Energy Price Information
draft-jennings-energy-pricing-01

Abstract

This specification defines media types for representing the future price of energy in JSON. It also defines a way for a client device, such as a car, refrigerator, air conditioner, water heater, or display to discover a web server that can provide the future price for local electrical energy. This will allow the client device to make intelligent decisions about when to use energy, and enable price distribution when the building is off-grid. It enables obtaining price from a local or non-local price server.
This draft is an early skeleton of a draft to start discussion around this idea.
The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet- Drafts is at http://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on January 11, 2012.

1. Overview

Many uses of energy can be shifted in time, or changed in quantity, based on price. Consider charging an electric car. For users that plug in cars at 9pm, they may not care when it actually charges, as long as it is ready at 8am when they need to go to work. This is a classic real time problem and can be optimized as long as the charger for the car has relevant information about how long it will take to charge and the cost of electricity between the current time and the time when the task needs to be complete.
Other devices such as refrigerators, air conditioners, and washers can similarly shift load. For their primary temperature regulation function, they can lower their setpoint (for cooling devices) when costs are low, and increase it when costs are high. The amount of deviation from the base target is keyed to the value of the price, operational considerations (e.g. not letting food freeze or spoil), or other non-price information available (e.g. occupancy). Devices such as displays (TV or computer) or lights can dim in some proportion to the electricity price, to balance cost and functionality. Devices with user-oriented time-outs (e.g. when an occupancy sensor's lack of seeing anyone in a space leads to a light going off) can adjust the length of such time-outs in proportion to price. Periodic functions (e.g. a refrigerator defrost cycle) can be shifted to the lowest cost time in the relevant time horizon. In general, the end-use device itself usually has the most knowledge about how best to act, and the the best access to internal actuators to accomplish the change.
Development around "Demand Response (DR)" has been advancing since around 2000. Most work in that area involves sending signals from the grid (DR-service provider) to a large building (commercial/industrial) or large device within it, to request load shedding or load shifting. There are then financial arrangements to pay the building owner for the service. More recently, the DR community and regulators have turned to enabling dynamic pricing so that the price customers actually pay at the meter more closely corresponds to the actual costs that the utility faces. Prices can be sent from the grid to an end use device, or from the grid to a gateway device (could be the meter) that then sends the prices to end use devices.
This specification defines a simple JSON[RFC4627]media type to provide the cost of energy at future points of time. It is an array of objects in which each object contains the time a new price will come into effect and the price at that time. JSON also defines a well known URL on a web server so that an HTTP client can retrieve this data. Finally as a way to automatically discover the web server, this specification defines a DHCP option to provide the host name of the web server.

At this time, only electricity is contemplated, but other resources do plausibly have time-varying prices, such as centrally provided steam or hot/cold water. Any resource (e.g. water) could use this mechanism to have a local price to distribute. Resources with a local supply constraint will then have a local price to ensure a balance with demand.

The base usage case for this specification is a time-varying electricity price with the current price and a set of future prices (confirmed or estimates), usually for a 24 hour period. This price comes from the electric utility. The price can be fetched directly from the utility. However, many alternate cases are also expected and supported. The building may have one entity (likely a piece of network equipment since it is always on already) that gets prices from the grid and all others get it from this building-local 'price server'. Both transactions use this mechanism.

The operator of the building may choose to present a higher price to devices in the building to take into account carbon emissions or other pollution from generating electricity. The building may also have local generation and/or storage, whose state and operation may indicate changes in price. For example, a building with an excess of solar power on-site may sell marginal electricity back to the grid at a low price. This would suggest lowering the price until supply and demand in the building were approximately in balance.

Some buildings operate off-grid, either all the time or intermittently. A building is a structure that uses resources and provides services. Common examples are homes, office, retail, and institutional buildings. Other building types include vehicles such as cars, ships, and airplanes. All these building types have electricity systems that would benefit from a price mechanism.

There are other protocols designed to get prices from the grid to a building, particularly to a building control system. One example of these is OpenADR. This mechanism complements rather than replaces these other mechanisms.

Electricity pricing has other aspects that complicate pricing. For example, in many places electricity use over a monthly billing period is sold in blocks, with the price increasing or decreasing with larger blocks depending on what the utility is trying to accomplish with the price. For example, the first five hundred kWh could be $0.10/kWh, the second 500 kWh $0.15, and so on. Thus, the monthly marginal price (what is paid if the consumption goes up or down modestly) is the last block used. This could be substantially different from an average price. There are many options for how utilities could combine blocks with dynamic prices. This specification is not attempting to provide a set of prices that are legally binding. Rather, it is intended to provide a simple and reasonably reliable set of prices that devices can use (when the alternative may be in fact no information at all).

Consider a typical residence with broadband Internet and a residential gateway that gets its IP address via DHCP from the service provider. The service provider would provide the domain of the local power

provider via DHCP. The residential gateway would get this and provide it in DHCP requests sent to the residential gateway. The residential gateway would also be able to override this, so if the consumer had arranged power from an alternative power provider, the name of that provider could be configured in the device.
A device on the residential network, such as a dishwasher, could find the energy provider name via DHCP. The dishwasher would then make an HTTP GET request to the well known URI defined in this specification. In other words, it would do an HTTP GET to the /.well_known/ electricity-price.json and would receive back an energyprice+json media type. For example

```
{
"currency" : "USD",
"prices":[
    { "time": "2011-04-12T23:20:00.00Z", "price": "0.028" },
    { "time": "2011-04-12T23:21:00.00Z", "price": "0.025" },
    { "time": "2011-04-12T23:22:00.00Z", "price": "0.021" }
]}
```

The above example shows a case where at 21:00 UTC, the price falls from 2.8 cents per KWh to 2.5 cents per kWh. Using kWh is fixed.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 *[RFC2119]*.

3. Semantics

Each media type carries a single JSON object that represents a set of prices and times. This object contains optional attributes described below and a mandatory array of one or more measurements.

validTill: Time at which this data series will become invalid. UTC time in RFC 3339 format.

currency: Optional. Specify currency in ISO 4217 [REF] currency code.

prices: Array of price objects. Mandatory and there must be at least one object in the array. Objects MUST be ordered in this array by time.

Each price time object contains several attributes, some of which are optional and some of which are mandatory.

time:
> Time this price becomes effective. UTC time in RFC 3339 format.

price: Price per kWh. The cost of energy changes to this price at the time in this object and remains at this price until the time of the next object in the prices array.

Open Issue: What is the best representation for time?
Open Issue: Is it OK that currency is optional?
Open Issue: How many entries can the array have? It would be nice to have some maximum size.
The price in the last entry in the series is ignored. That is, the purpose of the last entry is to close the time of the last period.
While 24 hours will be a typical time horizon, it could be shorter or longer.
Question: Can the request have a start time (zero for the present), so that if there is a limit on array size, one can get the rest?
Open Issue: should we be able to represent both buy and sell prices?

4. Well Known URL

A client that implements this specification uses the path "//.well-known/electricity-price.json" for the resource name unless the client has been configured with an alternative path.

5. DHCP

Open Issues: Is DHCP the best approach to discovery or would something else be better?

6. IANA Considerations

Note to RFC Editor: Please replace all occurrences of "RFC-AAAA" with the RFC number of this specification.

6.1. Well-Known URI Registration

IANA will make the following "Well Known URI" registration as described in RFC 5785:

URI suffix:	electricity-price.json
Change controller:	IETF <iesg@ietf.org>
Specification document(s):	[RFC-AAAA]
Related information:	None

6.2. DHCP Options

TBD

6.3. Media Type Registration

The following registrations are done following the procedure specified in [RFC4288] and [RFC3023].
Note to RFC Editor: Please replace all occurrences of "RFC-AAAA" with the RFC number of this specification.

6.3.1. energyprice+json Media Type Registration

TBD

7. Mapping to OpenADR

Lawrence Berkeley National Laboratory led the development of OpenADR initially (OpenADR v1.0), and it is now being formalized as an open standard through OASIS and national Smart Grid activity (OpenADR v2.0). At present, there are two relevant OASIS technical committees (TCs) that are relevant to the dynamic pricing (includes real-time prices) discussion: the Energy Interoperation TC (EI) and the Energy Market Information Exchange TC (EMIX). Each committee has a draft standard of the same name as the technical committee.
The OpenADR v2.0 standard will become a subset of what EI produces. EMIX is charged with defining a standard abstract form of price signaling. The details of how to represent a price product is defined in EMIX[EMIX] (then EI[EI] would reference and build implementation models, for e.g., XML schemas).
Both committees cover much more than just price (and price forecast) information. The discussion below focuses only on features relevant to this IETF specification. The OpenADR model uses XML as the data description language. OpenADR v1.0 and v2.0 can specify prices in different terms - absolute, multiple, or in relative terms to a base price (either additive or multiplicative).
Pricing can be a very complicated topic, but for the discussion here, we limit it to what this specification does- a schedule of time periods and a price for each period.
To represent time, EI and EMIX use WS-Calendar (also an OASIS standard), which provides for complex scheduling; simple price sequences use only a small part of this. Sequences are represented as a start time and a sequence of interval durations. As WS-Calendar builds on iCalendar (see RFC 5545) it uses the same date/time format as this draft.
A related issue is how to specify the current time to assure that the price source and user of the price have consistent time (or know how to adjust the schedule for a difference in time). This discussion does not consider this topic. So long as prices do not vary significantly from one time period to the next, and the time differences are not large, this issue is not of great concern.
EMIX can encode prices in several ways, including relative prices. For absolute prices, the price is simply a numeric value in cents/kWh for

the U.S. Other additional attributes relevant to price representations are under consideration (e.g., currency). The following is a sample excerpt of an OpenADR v1.0 price schedule:

```
<p:drEventData>
  <p:notificationTime>2009-06-02T17:15:00.0</p:notificationTime>
  <p:startTime>2009-06-03T00:00:00.0</p:startTime>
  <p:endTime>2009-06-03T23:59:00.0</p:endTime>
  <p:eventInfoInstances>
     <p:eventInfoTypeID>PRICE_ABSOLUTE</p:eventInfoTypeID>
     <p:eventInfoName>Price</p:eventInfoName>
     <p:eventInfoValues>
        <p:value>0.0</p:value>
        <p:timeOffset>0</p:timeOffset>
     </p:eventInfoValues>
     <p:eventInfoValues>
        <p:value>0.0</p:value>
        <p:timeOffset>3600</p:timeOffset>
     </p:eventInfoValues>
     ...
     <p:eventInfoValues>
        <p:value>0.0</p:value>
        <p:timeOffset>82800</p:timeOffset>
     </p:eventInfoValues>
  </p:eventInfoInstances>
</p:drEventData>
```

TBD - define a simple mapping to and from OpenADR.

8. Security Considerations

TBD
Further discussion of security proprieties for media types can be found in Section 6.3.

9. Privacy Considerations

TBD

10. Acknowledgement

We would like to thank Girish Ghatikar at LBNL for information and text about OpenADR. Thanks for helpful comments from many people including Scott Brim, <get your name here>.

11. References

11.1. Normative References

[RFC4627]

	Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.
[RFC3023]	Murata, M., St. Laurent, S. and D. Kohn, "XML Media Types", RFC 3023, January 2001.
[RFC4288]	Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

| **[EMIX]** | OASIS, "Energy Market Information Exchange (EMIX) Version 1.0, Committee Specification Draft 02 / Public Review Draft", April 2011. |
| **[EI]** | OASIS, "Energy Interoperation Version 1.0, Committee Specification Draft 01", November 2010. |

Authors' Addresses

Cullen Jennings Jennings Cisco 170 West Tasman Drive San Jose, CA 95134 USA Phone: +1 408 421-9990 EMail: fluffy@cisco.com

Bruce Nordman Nordman Lawrence Berkeley National Laboratory 1 Cyclotron Road Berkeley, CA 94720 USA EMail: BNordman@LBL.gov

Table of Contents