Efficient Demand Assignment in Multi-Connected Microgrids

Kirill Kogan  
University of Waterloo  
kirill.kogan@gmail.com

Srinivasan Keshav  
University of Waterloo  
keshav@uwaterloo.ca

Sergey Nikolenko  
Steklov Mathematical Institute  
sergey@logic.pdmi.ras.ru

Alejandro Lopez-Ortiz  
University of Waterloo  
alopez-o@uwaterloo.ca

ABSTRACT
With the proliferation of distributed generation, an electrical load can be satisfied either by a centralized generator or by local/nearby distributed generators. Given a set of resource demands in a collection of geographically co-located microgrids that are connected to the central grid and also potentially to each other, each such demand characterized by a power level and a duration, we study algorithms that allocate generation resources to the set of demands by configuring switched paths from sources to loads.

1. INTRODUCTION AND MOTIVATION
In recent years, electricity generation has been rapidly becoming more diverse: power is generated today not only from large, capital-intensive plants but also from numerous smaller-capacity resources including solar panels, wind turbines, and diesel gensets. This proliferation has made it possible for electric demands to be met with local generation, reducing distribution losses and simultaneously increasing energy security. Increasingly, sets of loads can rely nearly entirely on local generation resources, forming a microgrid, with access to the central grid used only as a backup.

We anticipate that in the future geographically-close microgrids will opportunistically form connections with each other to increase reliability. This would allow, for instance, a set of apartment complexes to augment their own diesel gensets with shared solar generation from a nearby office complex on weekends. This is a natural recapitulation of the self-organizing process by which electricity grids were formed in the first place, before centralized generation essentially eliminated micro-generation a century ago.

The focus of our work is on efficient demand satisfaction in the context of multi-connected microgrids, where a demand can be met by different generation resources: local, nearby, or on a regional context of multi-connected microgrids, where a demand can be met this way. Given a set of demands to a collection of geographically co-located microgrids that are connected to the central grid and also potentially to each other, each such demand characterized by a power level and a duration, we study algorithms that allocate generation resources to the set of demands by configuring switched paths from sources to loads.

With some simplifying assumptions, we find that the abstract problem of meeting time-limited loads (i.e., each load requires a certain power for a certain time) from a set of generation resources using a set of distributed switches is similar to the problem of assigning packets of a certain length arriving at the input ports of a rearrangeable optical switch to a set of output ports. Each packet corresponds to a demand, each input port to a generation resource, and each output port to a load. Given a set of demands, the minimum make-span assignment of these demands to loads is also the assignment that minimizes total delay, while the minimum set of configurations also minimizes switch wear and tear. We therefore extend past work in demand assignments in rearrangeable optical switches [2] to compute lower and upper bounds on the minimum number of rearrangements needed to meet a set of demands.

2. PROBLEM STATEMENT AND NOTATION
We make two simplifying assumptions in our work. First, we assume that all generators have the same cost of power production. Second, we assume that distribution losses are negligible.

We model a set of multi-connected microgrids with a switching system \((I, D)\) that consists of a set of inputs \(I\) (the generators) with port capacities \(c_i\) (the nominal power that they currently generate) and a set of demands \(D\) (elastic electrical loads) that are to be scheduled; a demand \(d\) is characterized by its length \(l(d)\) (how long the demand lasts), width \(w(d)\) (the power level of the demand), and a load balancing vector \(v(d)\) that contains the set of input ports available to process \(d\) (i.e., the set of generators that can feasibly meet this demand).

Time is discrete; we denote by \(L\) and \(l\) respectively the longest and shortest length in time slots among all given demands. If a demand \(d\) is assigned to input \(i\) at time \(t\), \(d\) uses \(w(d)\) bandwidth of port \(i\) during the time interval \([t, t + l(d) - 1]\). A schedule \(P\) is a sequence of configurations, where each configuration is a partial mapping of the demands to the inputs that has to satisfy constraints imposed by port capacities and load balancing vectors. The length of a configuration \(C\) is defined by the longest demand that is scheduled during \(C\). There is a non-negligible penalty, called configuration overhead, of \(V\) time slots between two consecutive configurations. Our goal is to satisfy loads in \(D\) as fast as possible. Note that the value of \(V\) can impact a scheduling decision. Therefore, we consider an additional objective: to minimize the total number of configurations.

The practically interesting case is one where each demand can be met from exactly two input ports, and one of them is shared among all demands. This situation arises naturally if local distribution networks, each covering its own region, are supplemented by a central grid. In this case, the problem is to reuse the central grid input in the most efficient manner in order to optimize either makespan or the number of configurations.

Copyright is held by the author/owner(s).
e-Energy'13, May 21–24, 2013, Berkeley, California, USA.
ACM 978-1-4503-2052-8/13/05.
Algorithm 1 Greedy Scheduling Policy (D, I)

1: \( D := D, C := \emptyset \)
2: while \( D \neq \emptyset \) do
3: \( \text{start new configuration} \ C := \emptyset, I' := I \)
4: \( \text{while there are available ports and demands do} \)
5: \( (i, d) := \text{Choose Port Demand}(D, I') \)
6: \( C := C \cup \{(i, d)\}, C'_i := C'_i - w(d), D := D \setminus \{d\} \)
7: \( \text{end while} \)
8: \( C := C \cup \{(C, D') \}, D := D \setminus \{d \mid d \in C\} \)
9: \( \text{end while} \)
10: \( \text{Return} \ C \).

Algorithm 2 SG

1: function \( \text{Choose Port Demand}(\{D_i\}, I) \)
2: for \( i := 2 \) to \( I \) do
3: if \( c_i > w(d) \) for some \( d \in D_i \) then
4: \( \text{return} \ (i, \text{Choose Demand}(D_i, c_i)) \)
5: \( \text{end if} \)
6: \( \text{end for} \)
7: \( \text{return} \ (1, \text{Choose First}(\{D_i\}, I)) \)
8: end function

Algorithm 3 SLD

1: function \( \text{Choose Demand}(D_i, c_i) \)
2: \( \text{return} \ \arg \max \{l(d) \mid d \in D_i\} \)
3: end function

Algorithm 4 SLP

1: function \( \text{Choose Demand}(D_i, c_i) \)
2: \( \text{return} \ \max \{l(d) \mid d \in D_i\} \)
3: end function

4. References