The physical infrastructure in (a) is represented logically in (b). Note that “Grid” represents all centralized generators available through the transmission network. The “distributed switch” refers to the set of switches in microgrids that determines how loads are matched to generators.

Motivation

- Local electricity generation already supplements centralized distribution in many developing countries with less-than-adequate central grids [1];
- In developed countries, local generation may still be cheaper, greener, or both;
- In the future, microgrids will opportunistically form connections with each other to increase reliability;
- This leads to the concept of “packetized electricity” [2, 3], where demands are represented as “data packets”, and power distribution becomes similar to packet scheduling in a reroutable switch [4];
- The world’s first electricity switch with packets of electricity has already been designed [5].

Our Goals and Contributions

- We consider efficient demand satisfaction in multi-connected microgrids, where a demand can be met by different generation resources;
- For each demand, a load balancing vector determines which generators are connected/available to satisfy it;
- In our present model, each load balancing vector allows the central grid plus one more generator;
- We concentrate on scheduling elastic non-preemptive demands [6]: demands that can be delayed for a while but cannot be preempted while servicing;
- Thus, we minimize the delay in satisfying a set of resource demands and the total number of configurations (reroutings in the switch);
- We build a systematic study of these online policies and explore the impact of various parameters of scheduling policies on objective functions.

Simplifying Assumptions

- All generators have the same cost of power;
- Distribution losses are negligible;
- There is a non-negligible penalty on switching called configuration overhead;
- Demands are elastic and non-preemptive, i.e., it is possible (though undesirable) to delay a demand;
- A demand cannot be split, it has to be satisfied from a single source;
- Each demand’s load balancing vector has a shared port (central grid) and one other port available for this demand.

Notation and Problem Statement

Given a switching system \((I, D)\):
- Each input has capacity \(c_i\);
- Each demand \(d\) has length \(l(d)\), width \(w(d)\), and load-balancing vector \(v(d)\);
- Time is slotted;
- A schedule \(P\) is a sequence of configurations; the length of a configuration \(C\) is defined by the longest demand that is scheduled during \(C\);
- There is a configuration overhead of \(V\) time slots between two consecutive configurations.

The objective is to satisfy loads in \(D\) as fast as possible, in terms of either the number of configurations or their total length.

Parameters of Scheduling Policies

Four important parameters define the behaviour of a scheduling policy:
- Input port capacities;
- Demand lengths;
- Demand widths;
- "Normalized load".

General Greedy Policies

\[ \text{GreedySchedulingPolicy}(D, I) \]
1. \( D := \emptyset, C := \emptyset \);
2. While \( D \neq \emptyset \) do
   3. Start new configuration \( C := \emptyset, I' := I \);
   4. While there are available ports and demands do
      5. \( (i, d) := \text{ChoosePortDemand}(D, I') \);
      6. \( C := C \cup \{(i, d)\}, c_i' := c_i' - w(d) \);
      7. \( D := D \setminus \{d\} \);
   8. Return \( C \).

\[ \text{SG (Shared Greedy)} \]
1. Function \( \text{ChoosePortDemand}(\{D\}_i, I) \)
   2. for \( i := 2 \) to \( \#d \) do
   3. if \( c_i > w(d) \) for some \( d \in D \), then
   4. return \( i, \text{ChooseFirst}(\{D\}, c_i) \);
5. Return \( 1, \text{ChooseFirst}(\{D\}_i, I) \).

Shared Longest Demand

\[ \text{SLD} \]
1. Function \( \text{ChooseDemand}(D_i, c_i) \)
2. Return \( \text{arg max}_d \{l(d) | d \in D_i\} \);
3. Function \( \text{ChooseFirst}(D = \{D\}_i, I) \)
4. Return \( \text{arg max}_{d \in D} \{l(d)\} \).

Shared Longest Port

\[ \text{SLP} \]
1. Function \( \text{ChooseDemand}(D_i, c_i) \)
2. Return \( \text{arg max}_d \{l(d) | d \in D_i\} \);
3. Function \( \text{ChooseFirst}(D = \{D\}_i, I) \)
4. Return \( \text{arg max}_d \{l(d) \} \).

Shared Best Product

\[ \text{SBP} \]
1. Function \( \text{ChooseDemand}(D_i, c_i) \)
2. Return \( \text{arg max}_d \{l(d) | d \in D_i\} \);
3. Function \( \text{ChooseFirst}(D = \{D\}_i, I) \)
4. Return \( \text{arg max}_d \{l(d) \} \).

Further Work

- A natural generalization to the case of arbitrary load balancing vectors;
- Scheduling in economic constraints, with costs entering the picture;
- More practical simulation study on real data (if such data for microgrids becomes available).

References