Electrical Power and Energy Systems 73 (2015) 157–162

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Dynamic intelligent load balancing in power distribution networks

Vladimir Zdraveski [a,*], Mirko Todorovski [b], Ljupco Kocarev [a,c,d]

[a] Faculty of Computer Science and Engineering, Ss. Cyril and Methodious University, 1000 Skopje, Macedonia
[b] Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodious University, 1000 Skopje, Macedonia
[c] Macedonian Academy of Sciences and Arts, Skopje, Macedonia
[d] BioCircuits Institute, University of California, San Diego, CA, USA

ARTICLE INFO

Article history:
Received 27 June 2014
Received in revised form 30 April 2015
Accepted 5 May 2015
Available online 19 May 2015

Keywords:
Power distribution network
Load balancing
Dynamic intelligent load balancing

ABSTRACT

Using hierarchical, client–server addressing concepts and identifying the load balancing problem among phases in 3-phase systems, we propose a novel, very simple algorithm for dynamic intelligent load balancing (DILB), that decreases the power losses in a power distribution network (PDN). Our solution is easily applicable to every part of a PDN, without essential changes to the last-line power installation. We present the DILB-extended PDN architecture, the algorithm itself as well as the results of the simulation on the well known IEEE 34-Bus, 37-Bus and 123-Bus networks that confirm the expected level of active power losses minimization.

Introduction

Power distribution networks are constantly being faced with an ever-growing load demand and/or could constantly experience distinct change from low/high to high/low load level. Although the loads are usually balanced across a three-phase distribution system during installation, the growth of the load demand as well as changes in load demand during a day, results into unbalanced state. Once a feeder is balanced it will initially be in balance but drift into unbalance as time goes on. The need for load balance and power loss minimization has triggered a vast variety of research on load balancing and load scheduling. There are two major phase balancing methods: feeder reconfiguration at the system level [1–3] and phase swapping at the feeder level.

Phase swapping has always been very active research topic. Several algorithms, including exhaustive search, backtracking algorithm, greedy algorithm, simulated annealing, genetic algorithm, and dynamic programming, applied to the phase balancing problem are recently discussed and compared by Wang et al. [4]. Here we give brief overview of the some of the work related to the phase swapping problem.

The authors of [5] developed a heuristic search algorithm for load balancing under constant and changing load, which was tested within the service area of Taipei South District, confirming both practicability of the algorithm and its positive impact. The phase swapping optimization schemes are proposed in [6]: the authors demonstrated the solution with an illustrative example and addressed balanced system improved characteristics. Simulated Annealing (SA) was introduced as an effective method to solve a power distribution phase balancing problem in [7]; the authors formulated the phase balancing problem as Mixed-Integer Programming (MIP) approach and then applied SA method, comparing results with several other algorithms and illustrating the global optimal solution calculation with an example feeder. The author of [8] developed a phase balancing algorithm with an emphasis to the practical aspects and implementation, which allows to indicate a limit on the number of phase move operations and exclusion of certain literals, difficult to access by the field crew. Three-phase balancing of distribution feeders using immune algorithm that guaranties fast convergence is proposed in [9], and applied in the distribution systems in Taiwan Power Company (Taipower) in order to address the feeder service tripping caused by the neutral over-current. The authors of [10] by utilizing the existing geographic information systems (GIS) features, presented a phase balancing application program that is based on an automated mapping/facilities management/geographic information system (AM/FM/GIS). Solving the phase balancing problem, the authors of [11] presented a new combinational method, based on bacterial foraging algorithm (BF-NM), and achieved quick access to the best solution and high convergence.

Periodically crews rebalance feeders, during periods of maintenance or restoration, when a new customer is to be connected or if the percentage of unbalance exceeds some number and the phase balance for existing feeders has become significantly unbalanced.

* Corresponding author at: Ruger Boskovik 16, 1000 Skopje, Macedonia. Tel.: +389 70428072.
E-mail address: vladimir.zdraveski@finki.ukim.mk (V. Zdraveski).

http://dx.doi.org/10.1016/j.ijepes.2015.05.012
0142-0615/

Three factors are considered in making a decision to re-balance a feeder: the monetary cost of making the tap change(s), the expected increase in feeder balance and the temporary interruption of power to the customer. Current phase rebalance is complicated to be maintained manually with the old power meters that have no option for embedding intelligence or connecting them in a logical (computer) network and is, therefore, unpractical, time consuming, and costly.

In this paper we take an advantage of new digital power meters that have computational and communication capabilities and are placed in the installation panels, some of which already connected to a local computer network or even further maintained via Internet. The network of smart power meters is a good basis for incorporating additional intelligence and advanced dynamic algorithms that will enable a more stable and economic PDN. Identifying the load balancing problem among phases in 3-phase systems and utilizing the existing smart network of power meters, we propose a simple novel algorithm for dynamic, intelligent load balancing (DILB). Some parts of the proposed architecture are based on the existing hierarchical, client–server addressing concepts. We present both the algorithm and the DILB-extended PDN architecture, which are easily applicable to every part of the power distribution network (PDN), without the need for essential changes in the last-line power installation. The DILB algorithm decreases the active power losses in a PDN, which has been demonstrated via simulation with well-known IEEE 34-Bus, 37-Bus, and 123-Bus networks, so that the expected level of active power losses minimization was confirmed.

The outline of the paper is as follows: Section 'Architecture description' explores the main idea of the concept and the algorithm itself is defined in section 'Optimization algorithm overview'. The real data measurements and simulation results are presented in section 'Measurements and simulation results'. Section 'DILB smart distribution network' describes the architecture extension of an existing PDN that is simulated on the IEEE 34-Bus, 37-Bus and 123-Bus networks. Section 'Conclusions' addresses the main benefits of the proposed concept.

Architecture description

In order to show the basic idea of our novel concept, we will simplify the power distribution network to a small size tree consisting of one parent node and n child nodes, shown in Fig. 1. It is actually a small part of a real PDN, so no further constraints will be defined, except the small number of nodes. If we would like to describe the last-line of a power installation, then the node N_0, may be considered as a central power meter in a building and nodes N_1 to Nn as power meters in each apartment. If we prefer to go a level up to a more central part of the PDN, then we may consider the node N_0 as a substation and nodes from N_1 to N_n as buildings. It should be pointed out that in the paper the power factors of loads are considered to be equal.

What we propose is to add an intelligent rotary cam switch in each node (N_0 to N_n) and also a microcontroller that will drive the rotary cam switch. Moreover, controllers will be interconnected using some of the widely used communication technologies, such as WiFi, Bluetooth or modulated signals through the PDN wires.

Each microcontroller will continuously measure the current and voltage (in order to calculate the power) in its node with a predefined sampling rate. Then all nodes will synchronize to each other and move the rotary cam switches in positions which will result in a maximal load balance in the root node (N_0 in our case).

In a more optimal scenario (for a fast initial implementation and a proof of concept), we may actually remove the root node N_0 and let the nodes N_1 to N_n form an ad hoc network and do the synchronization themselves. In both scenarios, we should pay attention to the scale of the network, thus providing sufficient communication channel bandwidth and processing power in the node-controllers.

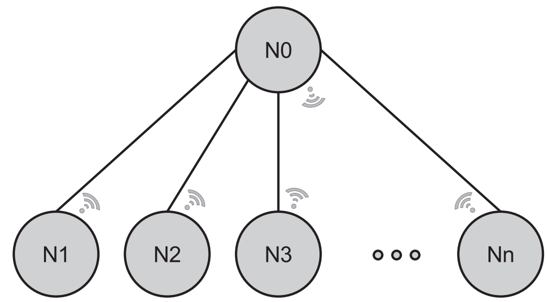

Fig. 1. Simplified PDN architecture formed by only one parent node and its children.

Targeting a more scalable and enterprise architecture, we may implement a level abstraction paradigm and use the simplified network from Fig. 1 as a building block of a more complex PDN. Following the concept of a network of networks, the result will be a network of the same type as the one in Fig. 1, except that nodes (N_1 to N_n) will also be networks. The process may be recursively extended, resulting in a large network with a generic topology shown in Fig. 2.

Note the border nodes (N_b), that have two communication interfaces, control the level abstraction, so the node N_0 in Fig. 2 will not be aware of the nodes N_1 to N_n, in both left and right subtree. Also, note that nodes with the same ID's may exist in different private networks, which is an addressing scheme similar to the IP addressing in computer networks and the network address translation (NAT) concept. Nodes marked as N_b, the border nodes, will get their ID dynamically from the parent node (N_0 in our case). The concept is scalable, so N_0 may also be a border node in another larger PDN and so on.

Optimization algorithm overview

Every node will calculate the power value for each phase, thus we define P_j, the vector of power values by phase at the node N_j, as

$$P_j = [P_{rj} \quad P_{sj} \quad P_{tj}]^T, \tag{1}$$

where T denotes the transpose. We arrange all vectors in a matrix:

$$P_{rst} = \begin{bmatrix} P_{r1} & P_{r2} & & P_{rn} \\ P_{s1} & P_{s2} & \dots & P_{sn} \\ P_{t1} & P_{t2} & & P_{tn} \end{bmatrix}. \tag{2}$$

The main goal in achieving the optimal load balance is to minimize the difference among loads by phase, d, i.e. to minimize the expression:

$$d = 100 * \frac{|P_r - P_s| + |P_s - P_t| + |P_t - P_r|}{2 * (P_r + P_s + P_t)}, \tag{3}$$

where

$$P_r = \sum_{j=1}^{n} P_{rj} \tag{4}$$

stands for the row sum of the first row of the matrix P_{rst} which is the total load of the phase r measured in the node N_0. Similarly, P_s and P_t stand for the total load by $s$and t, respectively, and are defined as

$$P_s = \sum_{j=1}^{n} P_{sj} \quad \text{and} \quad P_t = \sum_{j=1}^{n} P_{tj}.$$