Data Re-sequencing in Smart Grids

Philani Khumalo[1]
Department of Electronic Engineering
DUT
Durban, KZN
philanipk@gmail.com

Bakhe Nleya[2]
Department of Electronic Engineering
DUT
Durban, KZN

***Abstract*— Currently, legacy electrical power grids are being modernized into Smart Grids. These will in turn play a crucial role in real-time balancing between energy productions versus energy consumption. Each Smart Grids will dedicate an advanced metering infrastructure that facilitates collection, storing as well as analyzing data from smart meters to the authorized parties, and also carrying commands, requests, messages and software up-dates from the authorized parties to the smart meters. As such, data aggregation as well as unimpeded data relaying is a prerequisite for guaranteeing a large acceptance and deployment of Smart Grids. In this paper we provide an overview framework for analyzing packet re-sequencing within the Smart Grid. We utilize the random shortest path calculation algorithm to select the desired routes from source to a given destination. It is from among these that ultimately multipath (dual path) routing of the Advanced Metering Infrastructure data is car-ried out, hence resulting in re-sequencing necessities.**

Keywords— smart metering, advanced metering infrastructure (AMI), algorithms, communication networks, re-sequencing delays.

I. INTRODUCTION

The traditional power grids are being modernized and reengineered by way of exploiting Information and Communication Technologies (ICT) into Smart Grids (SGs), and as such SGs are able to monitor, protect, and optimize the operation of all entities within the SG. The primary goal of this innovation is to promote energy distribution efficiency, reliability, sustainability and integration of renewable sources. A typical SG incorporates several layers (figure 1) that include; a power system layer that encompasses distributed power generation, transmission, distribution as well as consumer systems; a power control layer, that monitors and controls the entire smart grid; a communication layer, which facilitates semi du-plex data exchange within the smart grid environment; a security layer, which ensures data integrity, confidentiality, au-thentication as well as availability; and finally an application layer, which facilitates various innovative smart grid applications to power users and utilities, a key example being Advanced Metering Infrastructure (AMI). The communication layer is one of the most critical elements that enables smart grid applications. Various entities/functionalities contribute to overall data that is exchanged within the SG. Examples include:

The Phasor Measurement Unit: which dedicates towards providing synchronized phasor measurements of key quantizes such as voltages and currents in the power grid. It does so by way of sampling these quantities waveforms using a com-mon synchronizing sampling signal tapped from Global Positioning Satellite (GPS) system. Typically the reporting is fixed at 20 to 50Hz. Each power utility will generally have a centralized , Phasor Data Concentrator (PDC) for aggregating and aligning data from several PMUs within the SG before relaying it to a Central Facility where synchronization network wide is accomplished.

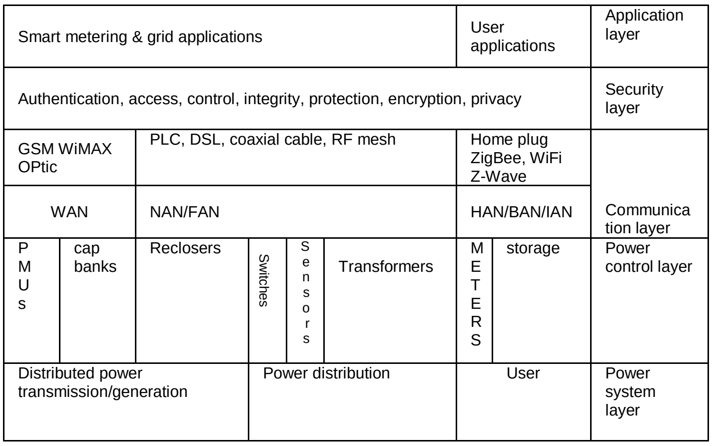

Fig. 1. Smart Grid Multi-layer [1]

Standard IEEE Std C37.118-2005 defines four message types for PMUs output as data, configuration, header, and command [2]. Typically each message is 100~200 bytes. Overall transmission latency should is bounded to 10~20 microsec-onds or less [3].

An Advance Metering Infrastructure (AMI) service: which is essentially an amalgamation of automatic meter reading systems (AMR) in the distribution subsystem, a two way communication to manage consumption demands as well as several specific actuators collectively generate lots of data within the SG. AMI is capable affording SG customers real-time pricing, peak shaving as well as energy conservation. Typically consumption status reporting/surveillance is carried out 4~6 times an hour. Primarily AMI message types include: Data, Configuration, Header, and OutCommand [3]. Data message comprises current consumption from customers, device ID, meter reading, time stamps, and other relevant information. Configuration message contains the AMI's configuration information. OutCommand message broadcasts real-time pricing to users.

Table 1. AMI message types

Type	Sink	Destination	Length
data	AMI	controller	<100
Configuration	AMI	controller	100~200
OutCommand	AMI	customer devices	<100
Header	AMI	controller	<50
InCommand	controller	AMI	100~200

Typical AMI latency requirement is about 1 second [3]. Header: A human readable message. InCommand message relays commands from supervisor con-troller.

Table 2 Maximum bound communication requirements

	Components	Lengths(bits)	Latency (ms)	BW(Mbps)
uplink	PMU	10x2000	10	2
	AMI	10000x1000	1000	10
	miscellaneous	100x1000	10	10
	SUM			22
downlink	controclentre	(10+100) x1000	10	11
		10,000x1000	1000	10
	SUM			21

An AMI application network should meet certain communica-tion constraints to achieve real-time feedback control of power demand and usage so as to promote efficiency in both demand distribution by the utility and usage [1]. These are summarized as follows [2]:

- *Reliability:* The AMI network must guarantee the arrival of each AMI meter reading and command mes-sages.
- *Scalability:* The designed network for AMI should be able to provide support to large numbers of AMI nodes. This requires the provisioning of sufficient enabling transmission bandwidth supporting desired QoS for the AMI.
- *Real time communication:* The round trip should be desirably short enough to in order fulfil the real-time control and feedback requirements.
- *Sequencing*. The data packets should be time stamped to guarantee the order of the data packets in the receiv-ing base.

The various SG applications, are very diverse in terms of network requirements e.g. payload sizes, data sampling requirements, loss tolerance, and latency. Latency refers to the amount of time a data packet takes to from one point (sender) on the network to another (receiver). Note however that latency has four main contributing factors, namely:

- Processing delay, T_{proc}: the delays by operations such as medium access adaptation, coding /decoding, switch fabric route configuring, routing, message authentications codes generation /line coding.
- Propagation delay, T_{prop}: this delay depends on the permittivity of the transmission medium as well as distance propagated by the signal in the same medium.
- Transmission delay, t_{tx}: this time required to transmit the data and defined as the ratio of the data size to the link speed. Queuing delay, T_q: the time spent by data packets awaiting, transmission or switching and is dependant of service discipline.

It is imperative that the SG communication infrastructure have tight bounds on its latency characteristics as it is one of the most stringent requirements for the grid. Adequate bandwidth provisioning is another important network requirement. This is an important criteria used in determining the transmission media to be relied upon for the SG communication because of the massive number of endpoints.

Table 3. Latency requirements [3]

Application	**Origin of data/place data is required**	**Latency Requirement**	***Date time window***
Station estimation	All substations/control center	1 second	*instant*
Transient stability	Generating substations/Application server	100ms	*10-50 Cycles(167ms-830ms*
Small signal stability	Some key locations/application server	1second	*minutes*
Voltage stability	Some key locations/application server	1-5seconds	*minutes*
Post mortem analysis	*All PMU And digital fault recorder Data/historian*	*N/A*	*Instant and Event data*

The bandwidth requirements can quickly become untenable if appropriate precautions are not taken and that would lead to increases in latency.

II. COMMUNICATION INFRISTRUCTURE

SGs are being rolled out in the form of intercon-nected microgrids. A microgrid as such is considered to be the primary building block of future smart grids [4]. It is a cluster of power energy generation systems, storages, control systems and loads which under normal operation would connect with a traditional power grid (Macrogrid) so as to facilitate power transfer between the two should the need arise. Users within a microgrid domain can generate low voltage electrical power using distributed generation, such as wind turbines, solar pan-els ,bio fuels and fuel cells. The connection point with the Macrogrid can be disconnected, thus enabling the microgrid operating autonomously [5]. It can now be regarded as islanded, in which users rely only on power generated within it. In this way, the islanded microgrid's ability to isolate itself from the rest of the power network leads to a reliable power supply within it, i.e. comparatively, this intentional islanding provide a higher local reliability than that provided by the power system as a whole [6].

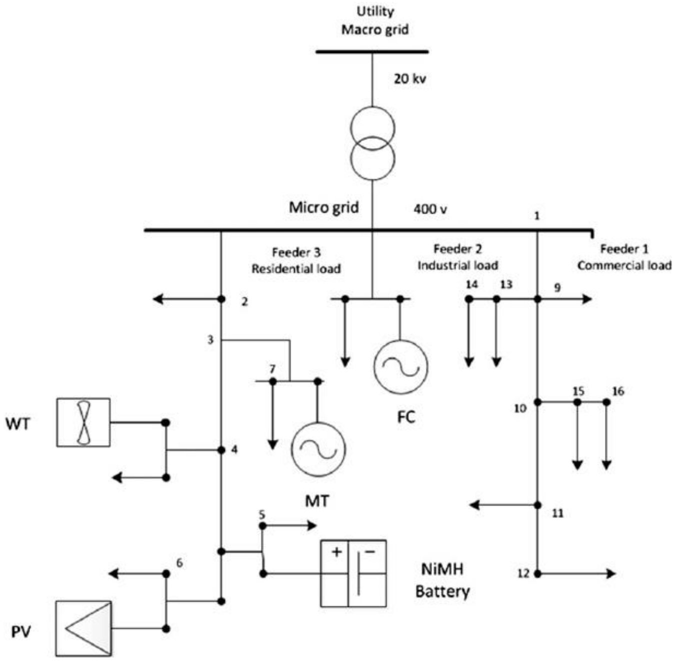

Figure 2. A Microgrid typical model [6].

From a communications network infrastructure perspective, the interconnected microgrids (clusters) form a distributed communications network.

To facilitate inter-microgrid communications, each micro-grid has a designated communications node(MCC) that man-ages key network related resources information, such as e.g., static nodes and link information, candidate routes for all destinations, resources state (available wavelengths) for each outgoing/output link(s),and exchanged link resource informa-tion from other nodes [7].

In this paper we assume that the SG is build in the manner outlined in section II, and therefore assume all the SG applications including AMI will rely on the MCCs for communications throughout the entire SG. the MCC normally calculates candidate routes as well as their respective sustain-able transmission rates (μ_i) in advance and selects an appropriate route when a request arises. By default all minimum hop routes are the candidate routes to the destination node. However given the numerous communicating points in a SG, a key challenge is find the distances between all pairs of communication nodes taking account their weights. The weight of each path can be determined by the number of nodes to be traversed, current traffic intensity on the path, as well as blocking probability etc. The entire web of nodes can be modeled as a weighted directed graph. Our task will be to devise algorithms to fast compute all possible distances between all pairs of vertices on that graph, with an ultimate goal of finding the shortest path from any given vertex (source) to another (destination). Initially a matrix of all possible distances of a given graph and then compute its distance product [7] [8]. A distance product of a matrix is defined by letting *A* bean *n* x *m* matrix and *B* bean *m* x *n* matrix. Their distance product *A* x *B* =C would be of size *n* x *n* such that

$c_{ij} = min_{k=1}^{m}\{a_{ik} + b_{kj}\}\ for\ \ 1 \le i,j \ge n$

In order to construct shortest paths, the notion of witnesses is used. The matrix n x n $=W$ is a witness matrix for the distance product C =A*B provided for every $1 \le i,j \le n$ we have $1 \le w_{ij} \le m$ and $c_{ij} = a_i w_{ij} + b_{w_{ij}}$

To find the shortest path, we can use the random shortest path algorithm which computes shortest path between all pairs of vertices of a directed graph of n vertices in which all edge weights are taken from a set .{-*M* ,...,*0*,..., *M*}. The algorithm receives an *n* x *n* matrix *D* containing the weights (lengths) of the edges of the directed graph (representing the SG network). The vertex set of the graph is V={1, 2,..*n*}, where each member The weight of the directed edge from *i* to *j* is d_{ij} if such an edge exists, otherwise it is +∞. The algorithm initially lets F→.D before running $\log_{3/2} n$ iterations each time settings $s \leftarrow 3/2^{l}$. A function rand next generates a subset *B* of *V*={1,2,...n.} whose individual elements are selected independently with a probability $p\alpha 1/s$. Matrices *F* [*.*B*].and *F* [*B**] where *F* [*B*] represents the matrix whose columns are the columns of *F* that correspond to the vertices of *B*, whilst *F* [*B**], is a matrix whose rows are the rows of *F* that correspond to the vertices of *B*. The distance product of matrices *F* [*.*B*].and *F* [*B**] is finally computed at the same time putting a limit *sM* on the absolute values of all entries. Finally a comparison of *F'* versus *F* is carried out. When an entry in *F* is smaller, it is copied to *F* as well as its corresponding witness copied to *W*. The algorithm is summarized as follows [8]:

$F \leftarrow D: W \leftarrow 0$
$M \leftarrow max\{|d_{ij}|: d_{ij} \ne \infty\}$
$for\ \ell \leftarrow 1\ to\ |\log_{3/2} n|do$
bergin
$s \leftarrow (3/2)^{\ell}$
$B \leftarrow rand(\{1,2,\dots,n\}, (9\ln n)/s)$
$(F', W) \leftarrow dist - prod(F[* B], F[B,*], sM$
for every $1 \le i,j \le n$ do
$if\ f'_{ij} < f_{ij} then f_{ij} \leftarrow f_{ij}, \omega_{ij} \leftarrow b_{\omega_{ij}} end\ if$
end
Return (*F*,*W*)

For any application the total delayT_d, will be computed from, $T_d = T_{prop} + T_{proc} + T_{tx} + T_q$. The queuing delay tqueue can easily become the biggest con-tributing factor to overall delay (latency) in a constrained network and hence we focus our attention to it.

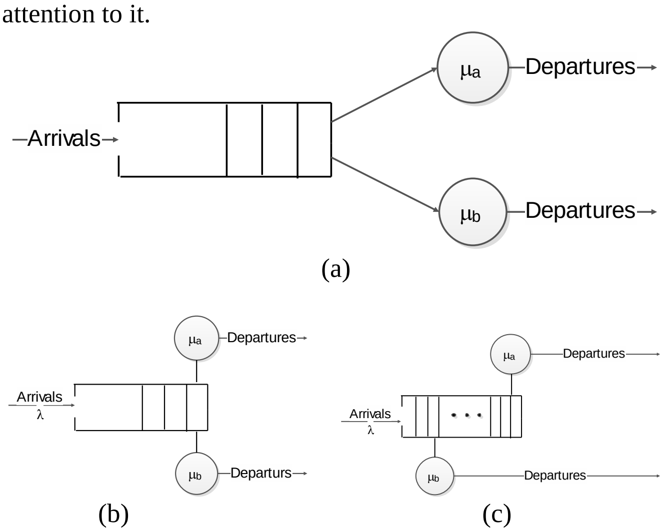

Figure 3. M/M/C servers

We assume multipath routing and focus on delays that are incurred at queuing buffers (MCCs). We make further as-sumptions as follows:

Poisson arrivals with rate λ. and exponential service time distributions with mean μ^{-1} are also assumed. The multipath outing strategy selects a set of randomly chosen shortest paths between source (s) to destination (d) For simplicity sake, we assume two paths with service rates μ_a and. μ_b ($\mu_a >. \mu_b$) are chosen and the service discipline is as follows:
A data packet arrives at the MCC to find:-

i. both paths a, and b idle: the packet is routed on path a since this path gives faster service on the average.
ii. both paths busy: the packet is queued in the buffer (figure 3(a).
iii. path a busy and path b idle: the newly arriving packet is queued in the buffer whose capacity is N and only if the number of packets in the buffer exceeds N will the packet be dispatched to path b .

Thus wherever path b becomes available, it starts serving a new arrival if and only if the number of waiting packets exceeds N. Once path a becomes available, it starts serving the packet at the head of the queue regardless of the total number of waiting packets. Note that path a is kept busy as much as possible since it has a faster transmission rate.
Note that the re-sequencing delay will depend on the position from which a packet is dispatched and therefore we can have up to N+1 policies. However we only consider two, defined as follows:
POLICY I: utilize the queuing discipline described, but with an added constraint that packets dispatched to the paths are always taken from the front end the queue (figure 3 b).
PLOCY II: utilize the same queuing discipline, but packets dispatched to path b are taken from the. $(N+1)^{th}$ position in the buffer, figure (3 c).

III. QUEING MODEL OVERVIEW

The two server queue can be modeled as a two-dimensional birth and death process.
We first define the state of the system at any given time as
$x = (x_0, x_1, x_2)$, where
x_0 = the number of packets in the queue.
$x_1 = 1$ or 0 depending on whether server j is busy or not.
The traffic intensity $\rho = \lambda/\mu_1 + \mu_2 \;\; for\; 0 \leq k \leq N$
Thus the expected number of packets in the system is:

$$E[n] = p(0,0.1) + \sum_{k=0}^{n} (k+1) p(k,1,0) + \sum_{k=0}^{x} (k+2)\, p(k,1,1)$$

The average queuing delay is
$T_Q = E[n]/\lambda$

Further if we let $\bar{x} = (\bar{x}_0, z)$., where z is an indicator whether the state is in sequence .(I) or out of sequence (O). A state would be would be in sequence either if server 2 is idle or if both servers are busy and the fast server1 is serving a packet that arrived in the system prior to the one being served on server 2 . Otherwise the state is out of sequence.

IV. PEFORMANCE EVALUATION

In this section we discuss the numerical results obtained from analytical expressions obtained in Section IV. We focus on the effect of system parameters and the policy selection on the queuing and re-sequencing delay. In this regard, the total delay (T_d) incurred is the sum of the queuing (T_q) as well as re-sequencing delays (T_{rd}) i.e.

$$T_d = T_q + T_{rd}$$

For policy I $T_d^I = T_q^I + T_{rd}^I$and for police II $T_d^{II} = T_q^{II} + T_{rd}^{II}$

V. DISCUSSION & CONLUSION

Our results show that the packet re-sequencing delay and the re-sequencing buffer occupancy drop when the traffic is spread over a larger number of homogeneous paths, although the network performance improvement quickly saturates when the number of paths increases. We find that the number of paths used in multipath routing should be small, say, up to three.

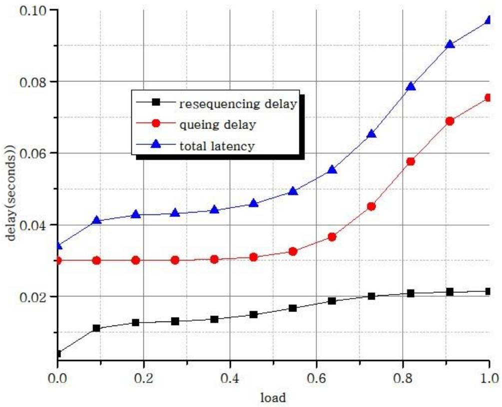

Figure 4. Delay versus workload for N = 0.

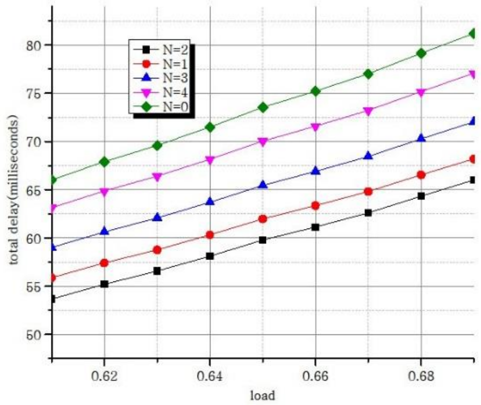

Figure 5. Total delay for various threshold settings (N).

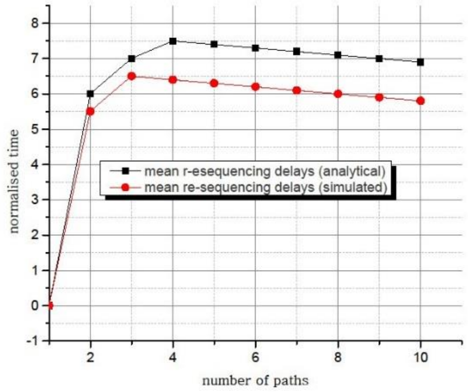

Figure 6. Total delay comparing the two policies.

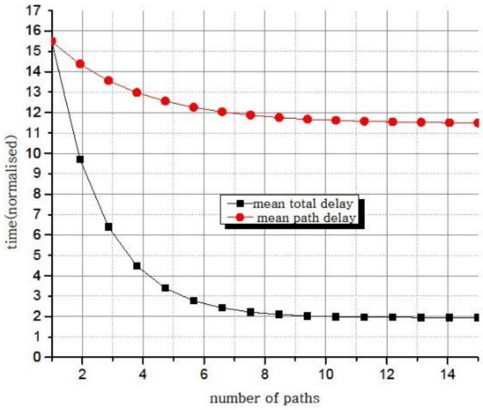

Figure 7.Packet delay versus number of multipaths.

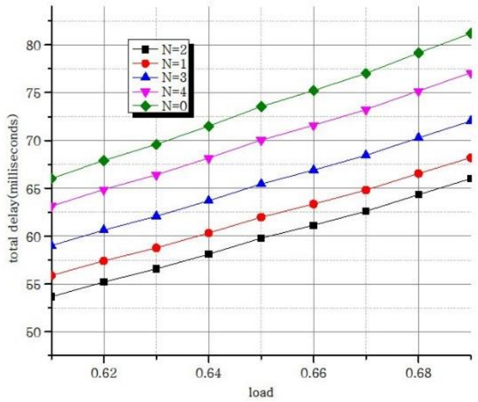

Figure 8.packet re-sequencing versus number of paths.

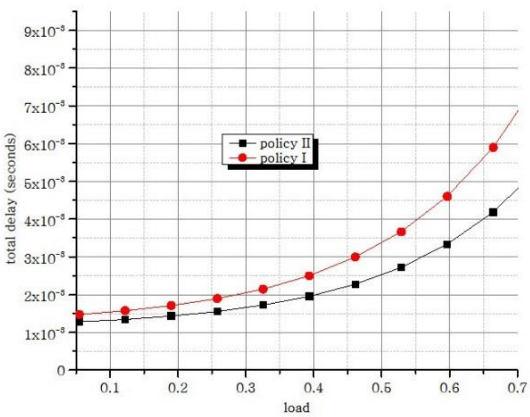

Figure 9.Re-sequencing buffer occupancy.

In figure 4, the total, queuing and re-sequencing delays are plotted against network traffic load. Recalling that the packets are always taken from the head of the queue, in this case the two policies perform identically. Next in figure 5, we sketch the total delay for various thresholds N for policy II only and we see that setting a value of N=2 and note that in this case minimal total delay is incurred. In the next step we investigate the effectiveness of multipath routing when different paths (with equal weights) are used. This is useful when the random shortest path algorithm intro-duced in section II, returns more than one shortest paths all with equal weights. So we assume that the routing weight to each of the set of shortest paths available is the same. We show the mean total delay, path delays, mean re-sequencing delays as well as the mean re-sequencing buffer occupancies. Fig. 7 shows the mean total delays as well as path delays in which both drop with increase in the number of paths. In figure 8, the mean re-sequencing delay, initially increases before dropping slightly as the number of paths is further in-creased, this attributed to by the reduction in variances of path delays when the number of paths is large. In figure 9, quality of service guarantees can be achieved, if large re-sequencing buffer sizes were to be used in cases where large numbers of paths is used. Finally, we conclude, by noting that, multipath routing can greatly enhance the overall performance of key applications.

REFERENCES

[1] Murat Kuzlu , Manisa Pipattanasomporn, Saifur Rahman. Communication network requirements for major smart grid applications in HAN, NAN and WAN. Computer Networks, Volume 67, pp 74-81.

[2] Yong-Hee Jeon. QoS Requirements for the Smart Grid Communications System. IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.3, March 2011.

[3] Prashant Kansal and Anjan Bose. Bandwidth and latency requirements for smart transmission grids. WSU ID: 11253840. available on the internet..

[4] Chao-Yang Lee and Chu-Sing Yang. Distributed Energy-Efficient Topology Control Algorithm in HomeM2MNetworks. International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation ,Volume 2012, Article ID 387192.

[5] L.F M de Moraes. Data Commmunication systems and their peformance. North Holland, 1987.

[6] Ramyar Rashed, Alan Fung, Farah Mohammadi. A survey on Advanced Metering Infrastructure. International Journal of Electrical Power & Energy Systems, Volume 63, December 2014, Pages 473–484.

[7] Communications Requirements Of Smart Grid Technologies. Department of Energy. October 5, 2010. http://energy.gov/sites/prod/files/gcprod/documents/Smart_Grid_Communications_Requirements_Report_10-05-2010.pdf.