Stability of a Cyber-Physical Smart Grid System using Cooperating Invariants.

Ashish Choudhari 1, Harini Ramaprasad 1, Tamal Paul 2, Jonathan W. Kimball 2, Maciej Zawodniok 2, Bruce McMillin 3, Sriram Chellappan 2

1Department of Electrical and Computer Engineering, Southern Illinois University Carbondale
2Department of Electrical and Computer Engineering, Missouri University of Science and Technology
3Department of Computer Science, Missouri University of Science and Technology

Abstract—
Cyber-Physical Systems (CPS) consist of computational components interconnected by computer networks that monitor and control switched physical entities interconnected by physical infrastructures. A fundamental challenge in the design and analysis of CPS is the lack of common semantics across the components. We address this challenge by employing a novel approach that composes the correctness of various components instead of their functionality using a conjunction of non-interfering logical invariants. We present a distributed algorithm that uses this approach to adaptively schedule power transfers between nodes in a smart power grid in such a way that the stability of both the computer network and the physical system are maintained. Simulation results demonstrate the necessity and usefulness of our approach in maintaining overall system stability in the presence of uncertainties in the computer network and with limited information about the global state of the system.

I. INTRODUCTION

A Cyber-Physical System (CPS) is one in which a set of networked, embedded computing (cyber) devices automatically monitor and control distributed physical entities in closed timing loops. Changes observed/effectected in the physical subsystem local to one cyber node are signalled to other relevant cyber nodes through messages over a computer network. A prime example of such a system is the smart power grid [23] where the overarching goal is to have embedded computing devices monitor and control distributed electric devices in order to achieve distributed generation, storage and transfer of power in a safe, reliable, efficient and secure manner.

Ensuring stability and correctness (both logical and temporal) of the system as a whole is a major challenge in CPS design. Any incorrectness or instability in one component can impact the same features of other components. For example, an action in the physical domain could affect the network domain and vice-versa, thus making correct scheduling of these actions paramount to overall system stability. The fundamental challenge in developing a design framework that unifies the various components is the heterogeneity of the component types, resulting in semantic gaps that must be bridged. For example, while the physical entities in a smart grid are electric devices whose stability and correctness may be expressed in terms of Lyapunov and Lyapunov-like functions, the notion of correctness in the context of the cyber devices are best expressed in the form of a conjunction of logical operators on system parameters.

Existing papers largely consider the stability of one or two components in isolation. For example, network delays affect system stability and considerable work focusses on determining system stability bounds as a function of injected delay [20]. Results from switched-systems theory [15] model the stability of the plant. Hybrid automata [19] and timed I/O automata [5] represent a simultaneous mix of continuous and discrete states in the verification process [11], [43]. Real-time scheduling is traditionally a function of a priori time bounds [29]. To consider components individually, or in pairs, requires that they be very stable such that the composition of the components into a CPS is stable.

In our work, we employ a fundamentally different approach that composes correctness instead of functionality. The basic idea, depicted in Figure 1, is to express the stability and correctness constraints of all components in the form of logical invariants and ensure that system actions are performed only if and when they are guaranteed not to violate the conjunction of these invariants. While this approach can be generalized to different cyber-physical systems and different functionalities therein, in the current paper, we focus on maintaining system stability in the presence of power transfers among nodes in a cyber-physical smart power grid.

The state of the physical system and, hence, its stability, is dependent on power transfers (series of power migrations) initiated by the cyber algorithm within each node in the system and by the state of the communication network that carries messages between the cyber nodes to signal initiation and acknowledgement of physical power migrations. The state and stability of the communication network is in turn affected by the number of migration messages in transit at any given time.

In this paper, we present a scheduling invariant that implements a distributed, adaptive algorithm for scheduling power migrations between nodes in a smart grid and demonstrate that a conjunction of such a scheduling invariant and an invariant for the physical system state is necessary to maintain overall system stability. In contrast to traditional real-time scheduling,
Correct scheduling in our context refers to initiating actions at appropriate times in a way that system stability is maintained rather than insisting that every action is initiated at a pre-defined time and must adhere to a pre-defined deadline.

The rest of this paper is organized as follows. Section II provides some background information and discusses related work. We present our system model and assumptions in Section III. Section IV presents our physical invariant and our adaptive scheduling scheme is presented in Section V. Section VI presents our resulting power management algorithm. Our simulation setup is introduced in Section VII and results are presented in Section VIII. Section IX presents a brief discussion and conclusions are presented in Section X.

II. BACKGROUND AND RELATED WORK

CPSs, with few exceptions, are switched dynamic systems. A switched system is a fundamentally continuous-time system with changes that occur at discrete times [28]. Analysis and design of CPSs is a challenge as any process must simultaneously take into account cyber, physical, and network aspects. Some work is breaking through this barrier. Acumen [48] bridges the gap between analytic models and simulation codes. Interface automata [12] checks for compatibility of components in composition by showing that they do not interfere with each other. Recent work [26] proposes a performance verification technique for CPS. This work assumes the use of a communication network such as CAN, FlexRay, etc. and use the relatively structured properties of the networking infrastructure to tightly bound network delays and, hence, control performance. Invariants and predicate transformers on the state of CPS was explored for dynamical systems in [41] and more recently as a formalism for invariant interaction and incremental invariant composition [7] and run-time assurance of operational modes [6]. The interaction of invariants for purely cyber processes has its origins in [31] which affords composition of sequential proofs governed by the property of noninterference. Recently, there have been several attempts to create comprehensive models for design and analysis of CPSs (such as [13], [8], [11]) that use domain-specific ontologies and hybrid systems techniques.

Correct scheduling of actions affecting one or more sub-components is key in such a CPS in order to maintain overall system stability. However, stability and scheduling are not a-priori, but must be adaptive based on events in the CPS. Mode-based real-time scheduling allows different modes of operation where different modes may have variation in their task set and/or task timing characteristics [39], [38], [18], [42], thereby allowing a degree of adaptation. However, existing approaches assume that mode parameters and mode change triggers are statically well-defined, allowing static analysis of individual modes and mode transitions, thus making them inapplicable in a CPS. Recent work proposes a technique for online reconfiguration of resource reservations using Constant Bandwidth Servers [27]. Elastic scheduling strategies [10] and feedback schedulers [45] allow for more dynamic adaptation, but adaptation is still typically performed at sporadic intervals, in contrast to the continuous adaptation needed in a CPS. Adaptive scheduling as in [14] dynamically changes the rates of task execution in response to system behavior, but would require complete abstraction of physical and network parameters, making its application to a CPS very challenging.

Scheduling of power demands for optimal energy management in a smart grid has been proposed [25]. However, this work only considers instantaneous power in the physical system and does not consider network behavior. Considering all continuous and discrete dynamics along with dynamic behavior simultaneously results in state explosion. While verification is possible, it is extremely challenging.

III. SYSTEM MODEL AND ASSUMPTIONS

Power Management Architecture. Figure 2 shows the architecture of a future generation smart grid (SmartGrid) [23]. The system is essentially a microgrid consisting of energy storage devices (DESD), energy resources (DRER) and LOADs. Each node is potentially owned and located in a residence or business and the basic idea is to share power among nodes in order to benefit the overall system. Intelligent flow controllers (nodes) contain Solid State Transformers (SSTs), that are physical actuators controlling power flow to and from a shared electrical bus under the direction of co-operating Distributed Grid Intelligence (DGI) processes.

The DGI processes are cyber algorithms that choose, negotiate and manage power transfers among nodes based on local information and information about the states of other nodes that is periodically exchanged among nodes. In the current work, we assume that all nodes are synchronized, for the sake of simplicity. Nodes periodically exchange state information with each other in a state collection phase. Next, a negotiation phase is performed in which nodes conduct negotiations to identify which power transfers need to be performed among nodes. The identified power transfers between pairs of nodes are then performed in a power transfer phase. One cycle is depicted in Figure 3, with one or more negotiation and power transfer phase pairs after a state collection phase. This entire
cycle of state collection, negotiation and power transfer phases is repeated.

**Fig. 2.** Smart Grid Power Management Architecture

**Power Transfer Model.** Power transfers within one phase are performed as a series of (periodic) power migrations, each transferring a given quantum (say $\delta$) of power. The cyber algorithm on the source (sender) node sends appropriate control signals to the local physical actuators to add a quantum of power to the electrical bus and sends a power migration message to the destination (receiver) node signalling this. Upon receiving this power migration message, the destination node sends control signals to its local physical actuators to remove a quantum of power from the electrical bus and sends an acknowledgement message to the source node.

**Physical System Model.** The physical system is a finite inertia microgrid, that is, a power system with a small number of generators and loads that acts independently of the grid. There is a relatively small rotating generator, whose inertia dominates the dynamics as described in the next section. There are some number of controllable nodes that participate in the power management by acting as either loads or generators. Nodes with excess generating capacity transfer power to nodes that have excess load.

**Communication Model.** The algorithm presented in this paper is independent of the topology of the communication infrastructure, being based instead on observed communication latencies at a given node. Hence, our algorithm may be applied to either a shared bus model or a network model with any given topology.

**IV. PHYSICAL SYSTEM ANALYSIS**

The physical system must satisfy one out of several criteria in order to be stable. The criteria are derived from an analysis of the continuous-time dynamics of the system, which are modeled as

$$\frac{d\omega}{dt} = -\frac{V_1 V_2}{J\omega X} \sin(\theta - \theta_0) - \frac{D}{J} (\omega - \omega_0) + \frac{P_{imb}}{J\omega} - \frac{kP^2}{J\omega},$$

where $\omega$ is the frequency, $\theta$ is the phase angle of the generator voltage, $\omega_0$ and $\theta_0$ are their nominal values, $P_{imb}$ is the net power imbalance due to outstanding messages, and the other terms are various physical parameters. The error energy, given by

$$V(\omega, \theta) = \frac{J}{2} (\omega - \omega_0)^2 + \frac{V_1 V_2}{\omega X} (1 - \cos(\theta - \theta_0)).$$

is a Lyapunov function (that is, a positive-definite function with a non-positive time derivative) if the system satisfies $I_{P_1}$, given by

$$I_{P_1} : (\omega - \omega_0)^2 (D\omega + m) + (\omega - \omega_0)(kP^2) > \delta K(\omega - \omega_0)$$

With other factors, $I_{P_1}$ ultimately imposes a limit on $\delta * K$, where $K$ is the quantum of power migrated with each message and $K$ is the number of outstanding messages.

In general, if a Lyapunov function exists for a particular physical system, then the system is stable. However, there are other conditions that also ensure stability for a switched system, which is a continuous-time system that is subject to external switching events. A Lyapunov-like function [9], [46], [47] is similar to a Lyapunov function in that it must be positive-definite, but its value may increase under some conditions. If the value of the Lyapunov-like function decreases at each switching event, then the system is stable. A final option is that the error may not decay to zero, but is bounded.

**A. Physical System Invariant ($I_P$)**

Combining the three conditions, a single invariant may be found,

$$I_P : I_{P_1} \lor (V(\omega, \theta) < V_{bound}) \lor (V(t) \leq V(t_x))$$

where $V_{bound}$ is the maximum allowable value of $V$, $V(t)$ is the value of $V(\omega, \theta)$ at the present time and $V(t_x)$ is its value at the most recent previous violation of $I_{P_1}$ due to a large value of $K$. Prior work [33], [32] has demonstrated that the system shown in Figure 8 is stable for certain combinations of steady-state power imbalance and droop constants in the SST controllers. The question at hand is whether the communication protocol ensures that the power imbalance does not exceed the allowable limit.
V. Adaptive Communication

As mentioned in the previous section, physical system stability at any given time depends on the total outstanding power in the electrical bus, i.e., on the product of the total number of outstanding messages, $K$, and the quantum of power migration, $\delta$. The value of $K$ at any given time also determines the stability of the communication network used to transfer power migration messages among nodes. $K$ is a function of the rate(s) of power migration of source node(s) and transfer times for messages over the communication network. In this section, we present an adaptive scheme for scheduling power migrations in such a way that $K$ does not exceed the stability limit of the network and $K \times \delta$ does not exceed the stability limit of the physical system.

Our approach is to 1) establish a limit on the maximum number of outstanding messages, $K_{\text{max,global}}$, in the system during a given power transfer phase based on network and physical system restrictions, 2) divide $K_{\text{max,global}}$ among communicating nodes so that each node $i$ is given a limit, $K_{\text{max},i}$, on the number of outstanding messages it can have in the system and 3) adaptively control a node’s message initiation rate to ensure that it does not have more than $K_{\text{max},i}$ outstanding messages at any given time.

A. Estimation of $K_{\text{max,global}}$

For a given power transfer phase, we assume that the number of outstanding messages from our nodes that the network can tolerate is bounded by $K_{\text{max,global}}$. In this paper, we assume that $K_{\text{max,global}}$ is known for a given power transfer phase and we focus on adaptive communication under the given bound. In practice, if a dedicated bus or network is used for the CPS, $K_{\text{max,global}}$ may be safely and tightly bounded. However, estimating $K_{\text{max,global}}$ over a large scale network like the Internet is non-trivial. Internet traffic can be bursty and it may happen that sudden increases in background traffic may cause dropped CPS packets (at routers), thereby compromising stability in the physical domain. Such unexpected dynamics and corresponding instability can be compensated by a conservative estimate of $K_{\text{max,global}}$ for a given power transfer phase based on profiling of Internet traffic during preceding phases, at the cost of decreased efficiency of the CPS. While a detailed discussion on the method for estimating $K_{\text{max,global}}$ is out of the scope of this paper, we provide a brief discussion in Section IX.

B. Distribution of $K_{\text{max,global}}$ among nodes

In general, $K_{\text{max,global}}$ may be distributed among nodes using any policy, with the simplest option being an equal distribution of $K_{\text{max,global}}$ among the $n$ nodes in the system. However, this may not be optimal in a situation where only a subset of nodes have excess power and, hence, are able to initiate power migrations. In this paper, since we assume that power transfers are initiated only at the end of a state collection phase, we use information about the states of nodes to distribute $K_{\text{max,global}}$ among nodes. Specifically, $K_{\text{max,global}}$ is distributed among the $n$ nodes in proportion to the excess power that nodes have at the end of the state collection phase, as shown in the equation below, where $P_s$ is the excess power of node $s$.

$$K_{\text{max},s} = \left[\left(\frac{P_s}{\sum_{i=1}^{n} P_i}\right) \times K_{\text{max,global}}\right]$$  \hspace{1cm} (5)

Note that, at the end of a state collection phase, every node has information about the power state of every other node. Hence, the above calculation of $K_{\text{max},s}$ may be performed independently by each node $s$.

C. Adaptive scheduling on each node

The goal of our algorithm is to adapt the rate $r_s$ - or, inversely, the period $p_s$ - of power migrations initiated by a node $s$ in response to observed system behavior while maintaining system stability. In other words, the goal is to adapt the period such that the number of outstanding messages of node $s$ at any given time, namely $K_s$ ($\sum_{i=1}^{n} K_i = K$), never exceeds its maximum allowed outstanding messages $K_{\text{max},s}$. Our algorithm uses the observed response times for power migration messages as a metric to judge the state of the network during a power transfer phase and as the trigger for period adaptation. The response time of a power migration message is defined as the time between the initiation of the message by its source node(s) and the time at which an acknowledgement for it is received from its destination node(s).

The basic approach of our algorithm is as follows. Every power migration message is assigned a relative deadline $D_s$ based on the current expected response time ($RT_s$) for messages initiated by node $s$. The expected response time is calculated as the average of a given number of previously observed response times. A “deadline miss” for any given message indicates longer network latencies than expected due to potential congestion in the network. In this situation, the algorithm decreases its expected response time by a predefined margin ($RT\text{Margin}$) and calculates a new, larger period and larger relative deadline for power migration messages in an effort to reduce congestion. In the worst case, this may result in a migration message being initiated only after acknowledgements for all previous messages have been received. If, on the other hand, acknowledgements are received earlier than expected for a given number, say $CtrMax$, of consecutive messages, the algorithm increases its expected response time by a predefined $RT\text{Margin}$ and calculates a new, smaller period and smaller relative deadline for power migrations that can still maintain network and physical system stability. Note that $RT\text{Margin}$ and $CtrMax$ are configurable system parameters that are assumed to be constant for all nodes in a given power transfer phase.

D. Scheduling invariant ($I_S$)

The approach outlined above results in the scheduling invariant shown in Equations 6 - 9.

$$I_S = I_k \land I_c \land I_p$$  \hspace{1cm} (6)
\[ I_k: K_s < K_{max_s} \]  
\[ I_c: RT_{s}^{ex} \leq PT - t \]  
\[ I_p: t - LT(s) \geq p_s \]

Here, \( PT \) is the end time of the power transfer phase, \( t \) is the time at which the invariant is evaluated and \( LT(s) \) is the time at which the last power migration message was initiated by node \( s \).

**E. Algorithm description**

We now describe the detailed working of our algorithm through various events. This description assumes that \( s \) is the source node and \( d \), the destination node of a given power transfer. Note that any re-calculation of node parameters affect only future power migration message initiations.

**Initialize Event:** This event is triggered at the beginning of each power transfer phase on every node \( s \) that acts as a source node in the given phase, i.e., on nodes with excess power that have successfully negotiated power transfers with nodes that have a demand for power. When this event is triggered, node \( s \) calculates \( K_{max_s} \) and the rate \( r_s \), period \( p_s \), and relative deadline \( D_s \) for messages to be initiated by it as shown in Equations 5, 10, 11 and 12, respectively. \( RT_{s}^{ex} \) is the average of response times observed during the preceding state collection or negotiation phases. A counter \( RTC_{tr} \) is initialized to zero. Fig. 4 pictorially represents the relationship between the various node parameters.

\[ r_s = \frac{(K_{max_s} - K_s)}{RT_{s}^{ex}} \]  
\[ p_s = \frac{1}{r_s} \]  
\[ D_s = RT_{s}^{ex} + RT_{Margin} \]

**Send Power Message Event:** This event is triggered (say, at time \( t \)) on a given node \( s \) to initiate a power migration to node \( d \), if the scheduling invariant is satisfied with the latest values of node parameters (\( K_s, RT_{s}^{ex}, p_s \)). When this event is triggered, control signals are sent to the physical system to add a quantum (\( \delta \)) of power to the electrical bus. A power migration message is created by \( s \), assigned an absolute deadline of \( t + D_s \), and sent over the network to node \( d \). \( K_s \) is incremented by 1 and a Deadline Miss event is created for time \( t + D_s \).

**Acknowledgment Received Event:** This event is triggered on a node \( s \) when it receives an acknowledgement from a node \( d \) for message \( m \). \( K_s \) is decremented by 1 and the response time \( RT \) of \( m \) is calculated. This event is handled in different ways if the acknowledgement is received before the deadline of the message it corresponds to (Case 1) and if it is received after the deadline (Case 2).

**Case 1:** The Deadline Miss event for message \( m \) is deleted from the list of future events. If \( RT < RT_{s}^{ex} - RT_{Margin} \), i.e., the acknowledgement has arrived earlier than expected, \( RTC_{tr} \) is incremented. If \( RTC_{tr} \) has reached \( CtrMax \), \( RT_{s}^{ex} \) is set equal to \( RT \) and node parameters are recalculated using Equations 10 - 12. Otherwise, \( RTC_{tr} \) is reset to zero. This situation is shown in Fig. 5.

**Case 2:** If the current value of \( RT_{s}^{ex} \) is less than \( RT \), \( RT_{s}^{ex} \) is incremented by \( RT_{Margin} \). If \( K_s < K_{max_s} \), node parameters are recalculated using Equations 10 - 12. Otherwise, \( r_s \) is set to \( 1/RT \) and other node parameters are calculated using Equations 11 and 12.

**Deadline Miss Event:** If an acknowledgement is received for a message before its deadline, its Deadline Miss event is deleted as noted above. Hence, the actual triggering of the Deadline Miss event indicates that the corresponding message has missed its deadline. In this situation, \( RTC_{tr} \) is reset to zero and \( RT_{s}^{ex} \) is incremented by \( RT_{Margin} \). If \( K_s < K_{max_s} \), node parameters are recalculated using Equations 10 - 12. Otherwise, \( r_s \) is set to \( 1/RT \) and other node parameters are calculated using Equations 11 and 12. Figure 6 depicts this situation.

**Receive Power Message Event:** This event is triggered when a destination node \( d \) receives a power migration message. When this event is triggered, control signals are sent to the physical system to remove a quantum (\( \delta \)) of power from the electrical bus and an acknowledgement is sent over the network to node \( s \).
VI. Resulting Power Management Algorithm

The DGI power balancing architecture of Section III is implemented by Algorithm 1, written in a communicating sequential processes (CSP)-like language [21], that executes commands operating on the local power system electronics. Its actions are asynchronous with respect to those of the power electronics and of the network. Thus, the potential exists for PowerBalance’s operation to interfere with the truth of the invariants $I_P$ and $I_S$.

In PowerBalance, each of $n$ co-operating processes (one on each node) executes an algorithm triggered by the state of the underlying power system, either high or low, which are the amount of variance above and below the nominal load. The goal of this algorithm is to allocate supply to demand, and as such, it approximates a distributed solution to the fractional knapsack problem [4]. As the algorithm executes, power migrations are initiated and the imbalance between the total amount of supply versus the total amount of demand decreases. The function $\text{migrate}(\delta, j)$ is a command to the underlying power system to provide/accept a quantum of power to/from a node $j$. The algorithm runs as many times as possible until a predetermined cycle is reached, denoting the end of a given power transfer phase, at which time other activities are scheduled.

For the proposed cyber-physical system, the final invariant is the conjunction of the scheduling invariant and the physical system stability invariant from the Lyapunov-like function. PowerBalance has the potential to interfere with power system Lyapunov function that governs the “mismatch” between the sending and receiving processes. To guarantee the CPS maintains the invariant, the system invariant is added as a guard, $I_P \land I_S$, as a weakest precondition on the communication, resulting in Algorithm 1. The values of $K_S$, $RT_s^{ex}$ and $p_s$, needed for evaluating $I_S$, are asynchronously calculated by the adaptive algorithm described in Section V and retrieved using the GetSchedulingParameters function shown in Algorithm 1.

Algorithm 1: PowerBalance Cyber Algorithm

```
PowerBalance $P_1 ::$

\begin{verbatim}
var k = 0
  do
    status = input()
    status = low → broadcast_request
    do
      ∀l = 1,⋯,n // Receive responses from any processes
      $P_l$?response[l] // Command the local device to transfer Power to $j$
    end
    $P_j$?select; // Send the Migration
    low = low + $\delta$
    $\text{migrate}(\delta, j)$ // Command the local device to transfer Power to $j$
    $\text{status} = high$ ∧ $P_j$?request → $P_j$?response
    $\text{status} = high$ ∧ $P_j$?select → // Receive the Winning Migration
    do
      $\text{migrate}(\delta, j)$ // Command the local device to receive Power from $j$
      high = high − $\delta$
    end
    k = k + 1
  end
\end{verbatim}
```

VII. Simulation Setup

Network Setup

We have implemented our algorithm for adaptive communication using OMNeT++, an open source network simulator [2], [44]. OMNeT++ is a discrete event simulator that provides an extensible, modular, component-based C++ simulation library. OMNeT++ supports wired, wireless and on-chip network simulation. Figure 7 shows the setup used to simulate every power transfer phase. Nodes (denoted as cpSNode), communicate with each other over links and routers (denoted as cpSSwitch) that employ first-in-first-out queues (denoted as abstractFifo) for storing and servicing incoming packets. The electrical bus to (from) which power is added (removed) by each cyber node is denoted by cpSGrid. In order to simulate background traffic in the network, we employ two special nodes, denoted as cpStrafficGen. Network link delays, queue service times and background traffic patterns are configurable. In the current paper, we focus on studying the behavior of our adaptive algorithm for given values of $K_{max,global}$ and no background traffic in the network. In future work, we propose to conduct a study of the effects of varying loads of background traffic on the estimation of $K_{max,global}$ and on the behavior of our adaptive algorithm.

For a given power transfer phase, the required input data for a given node (say $s$) in the system includes the power states of all nodes, the value of $K_{max,global}$ and the initial value of $RT_s^{ex}$. In addition, values of $RT_{Margin}$ and $CtrMax$ for all nodes are assumed to be provided as input. In the current set of simulations, the values of $RT_{Margin}$, $CtrMax$, network link delays and queue service times are assumed to be 0.25s, 2, 0.01s and 0.2s, respectively. In each simulation, one node has excess power (source node) and one node has demand for power (destination node). All results shown in this paper are
observations/measurements made at the source node.

Physical System Setup

The setup used for physical system simulation is shown in Figure 8. There are three controllable devices that represent solid-state transformers (SSTs) [23]. Each SST can generate or absorb active power from the grid. The larger grid is represented here by a single, lumped equivalent generator with finite inertia. That is, the system frequency is allowed to vary. If the net generation of the three SSTs increases, the system frequency increases. An increase in generation can result from an incomplete power migration. The system shown in Figure 8 was simulated with power imbalance determined through network simulation. The scale factor is such that each outstanding message corresponds to 200 kW of power imbalance. Previous work showed that steady-state imbalance of 2600 kW makes the physical system unstable [33], [32].

VIII. SIMULATION RESULTS

Simulations were conducted using only the scheduling invariant $I_S$ as a guard for power migrations. Figure 9 shows the results of a simulation where $K_{max\_global}$ is 20. Since there is only one node $s$ with excess power, $K_{max\_s}$ is also 20. When simulated, the number of outstanding messages rises to a maximum value of 19, and then settles around 10. Both the network (as measured by response time and period) and physical system (as measured by generator speed) are stable. Next, $K_{max\_global}$ was increased to 30, with all other network and physical parameters left unchanged. The result for this simulation is shown in Figure 10. The number of outstanding messages rises to 27 and then settles around 13. Once again, the network is stable due to the presence of $I_S$ as a guard. However, the physical system is unstable, as shown in Figure 10. Although the network timing parameters achieve steady-state, the generator speed oscillates and generally drifts upwards. Even after the power imbalance is resolved (around 19 s), the generator speed does not return to its nominal level of 377 rad/s. The network traffic shown in Figure 10 violates $I_P$, which was not used as a guard for this simulation.

One possible solution to the problem of network congestion is to increase the quantum of power transferred per message. Suppose the nominal power migration quantum is $\delta_{nom}$ for a nominal response time $RT_{nom}$. As the network state evolves, the response time will vary. In order to migrate the same amount of power, the quantum must vary as well, e.g.,

$$\delta_{act} = \delta_{nom} \times \frac{RT_{act}}{RT_{nom}}$$  \hspace{1cm} (13)
However, as the network traffic increases, $K$ will also increase. If $\delta \times K$ exceeds a given threshold, the physical system will become unstable.

Figure 11 illustrates the system response with some nominal value of $\delta$ (200 kW) corresponding to a nominal response time of 1.0 s. For this system, if $\delta K \leq 2600$ kW, the physical system is stable. For the case of Figure 11, $\delta K = 800$ kW in steady-state, so the system is stable as expected. If the response time doubles to 2.0 s, $\delta$ must also be doubled to 400 kW. As a result, $\delta K = 3600$ kW in steady-state as shown in Figure 12, and the physical system is unstable.

In all the above simulations, the instability in the physical system is reflected in a violation of $I_p$, demonstrating that, for the overall CPS to be stable, the conjunction of $I_S$ and $I_p$ must be used as a guard for power migrations, as shown in Algorithm 1.

**IX. DISCUSSION**

As mentioned in Section V-A, estimation of $K_{\text{max,global}}$ over a large scale network like the Internet is non-trivial. However, in the last two decades, extensive research has been conducted on modeling and predicting congestion and end-end delays over TCP/IP based Internet traffic [40], [17]. Typically, delays can be estimated based on a) latencies in TCP syn packets and their corresponding ack packets, which is a method accomplished at the starting epoch [24], [30]; b) latencies of multiple packets in flight, which is a method accomplished in the early (slow start) phase [40]; and c) latencies of packets in stream (or rather fluid flow models), which is a method accomplished in the congestion control phase [3], [22], [35]. Studies have shown all these methods can be accurate to a high degree [40] at corresponding phases. Other advanced methods for estimating delays in the Internet also exist. For instance, in [17], end-to-end delay measurement (from sources to destinations) are formulated as Bayesian inference problems and Markov chain Monte Carlo (MCMC) algorithms are designed to design to accurately predict delays based on prior measurements and models. More recently in [34], delays are estimated via cross correlation functions and correspondences between the sequence of a combination of both ack packets and that of data packets. While we do not specify a method for delay estimation of Internet based traffic, we believe that there are a number of well studied methods that will allow us to determine estimates of delays (trading off complexity, measurement time and accuracy) at various phases in order to derive $K_{\text{max,global}}$. One other
specific alternative we could consider is to let CPS nodes exploit the Explicit Congestion Notification (ECN) scheme, wherein packets are marked indicating impending congestion, instead of dropping them [16], [36], [37]. CPS end nodes could negotiate among themselves supporting ECN, and such a scheme can significantly improve system efficiency while also improving stability. How to accomplish this via an optimal selection of various parameters in a large scale network of many CPS nodes sending packets through diverse links is a open issue to investigate.

X. CONCLUSIONS

In this paper, we present a distributed, adaptive algorithm for scheduling power migrations among nodes in a cyber-physical smart power grid. The algorithm adapts the rate of power migrations initiated by a node in response to observed system behavior, with the goal of maintaining overall system stability in the presence of network uncertainties and limited information about the state of the global system. We employ a fundamentally new approach to achieve this goal that is based on composing the correctness of various system components, expressed as logical invariants, and employing the conjunction of these invariants as a guard for power migrations. Simulation results demonstrate the need and effectiveness of using the conjunction of invariants as a guard in maintaining system stability.

REFERENCES

[1] IEEE workshop on design, modeling and evaluation of cyber physical systems.


