Layers of Interacting Dynamic Networks: Motivation and Theory

Rahul Dhal

Abstract—We propose a framework for modeling two interacting network dynamics, that is promising as a representation of infrastructural dynamics modulated by underlying environmental uncertainties. Specifically, we propose using a stochastic automaton known as the influence model to capture the evolution of discrete-valued environmental uncertainty statuses. The environmental-uncertainty process is viewed as driving a linear network model that captures an infrastructural dynamics of interest. This two-layer hybrid network model is shown to permit low-computation statistical (moment) analysis of meshed environmental and infrastructure dynamics.

I. INTRODUCTION

Infrastructure networks such as the United States National Airspace System (NAS) and the power grid are increasingly being forced to operate near their limits, and yet are expected to achieve high efficiencies to reduce costs. At the same time, these networks are becoming increasingly sophisticated and multi-faceted: for instance, traffic in the NAS is becoming increasingly heterogeneous (e.g., including unmanned systems), and is subject to new environmental-impact- and security-based restrictions. As a result, there is a growing need for cyber-decision-support tools that assist human operators in managing the infrastructure dynamics, at a full-network scale and multi-hour time horizon. Unfortunately, at this scale and horizon, uncertain environmental factors (e.g., convective weather in the NAS, wind speeds and cloud cover as they impact renewable generation) often have a significant impact on system operation. Decision-support tools must account for these environmental uncertainties, see e.g. [1].

A critical first step towards building strategic decision support tools is to develop tractable unified models that capture relevant dynamics of the infrastructure network along with the uncertain environmental-impact dynamics, which are themselves governed by a networked stochastic process with complex spatio-temporal characteristics. This meshed modeling goal is challenging: two different network dynamics, which are interfaced in a complex way, must be captured. Further, the two dynamics that must be modeled are varied—the infrastructure-network’s dynamics are typically deterministic and often continuous-valued, while the environmental-impact dynamics are stochastic and sometimes discrete-valued (e.g., reflecting presence or absence of convective weather, or cloudiness vs. clear sky, for instance). A unifying model thus must be flexible enough to capture complex hybrid interactions between two different networked processes, which may be stochastic. At the same time, we require models that are simple enough to permit 1) real-time statistical characterization of a large-scale network (perhaps involving hundreds to thousands of components), and 2) management strategy design.

The goal of this paper is to introduce a class of layered network models which are promising for representing meshed environmental-uncertainty and infrastructure-network dynamics. The discrete-time model that we study comprises two layers. Specifically, a quasi-linear stochastic automaton known as the influence model is used to capture networked propagation of discrete-valued environmental-impact variables. This environmental-impact model is viewed as driving a deterministic linear network dynamics, which represents or approximates the relevant infrastructure-network dynamics. This two-layered network model is illustrated in Figure-1.

The layered architecture that we introduce in this work builds on our earlier efforts to model stochastic spatio-temporal propagation of weather using the influence model, for air traffic management [1], [2], [3]. In our recent work [4], we have extended the influence-model paradigm to a layered model, to approximate networked convective-weather propagation and consequent traffic congestion at a bottleneck queue in an air traffic system. The model is a specific instance of the layered network models developed here, in which the infrastructure-network dynamics is a scalar process (since only one queue is represented). The methodologies introduced here hold promise to extend the air-traffic-system analysis introduced in [4], and to inform modeling of future power-systems with significant renewable generation as well as disease-spread dynamics. In this article, we focus on introducing the two-layer model in generality, and describing basic statistical (moment) analyses of the meshed dynamics. We also briefly review the air transportation application, to illustrate the model.

We envision using the layered network model to characterize networks which are operating in the face of environmental uncertainties (which is themselves a networked process).
Such interfaced dynamical systems are often studied using numerical techniques (e.g., Monte Carlo analysis). However, analytical tools are needed, to identify important relationships between the networks’ topologies and their dynamics, and to facilitate their design. We believe that the layered network model is flexible enough to capture a broad range of such complex interactions, yet has enough structure to permit meaningful analysis.

The remainder of the article is organized as follows. Background on the influence model, and on notations for certain Kronecker-product, is given in Section II. We point out a couple of applications of the influence model beyond modeling stochastic uncertainties, in Section II-B. In Section III we present the model. In Section IV we present some preliminary results on the model and discuss some qualitative results on the model.

II. BACKGROUND AND NOTATION

In this section, we review mathematical notations and modeling constructs that are needed to formulate and analyze the layered network model. Specifically, we first briefly review a vector operator called the block Kronecker product, that turns out to be useful for capturing interfaced local- and network- dynamics in both layers of the network model. We then review a quasi-linear stochastic automaton network model known as the influence model [5,6], which we will use to capture environmental uncertainty propagation.

A. Block Kronecker Product

Consider two vectors $X$ and $Y$ that are partitioned as follows.

\[
X^T = \begin{bmatrix} x_1^T & x_2^T & \ldots & x_m^T \end{bmatrix}, \quad Y^T = \begin{bmatrix} y_1^T & y_2^T & \ldots & y_n^T \end{bmatrix},
\]

where $m$ and $n$ are positive integers. The block Kronecker product of the vectors $X$ and $Y$ is defined as

\[
X \otimes Y = \begin{bmatrix} x_1 \otimes y_1 & \cdots & \cdots & x_m \otimes y_n \end{bmatrix},
\]

The block Kronecker product is a permutation of the usual Kronecker product for two vectors, and is useful for maintaining component substructures in network-modeling applications. For more details on the block Kronecker product, including the specification of the permutation matrix that yields the regular Kronecker product, see [7].

B. Influence Model

An influence model is a particular quasi-linear stochastic automaton network model, that can be viewed as a network of interacting discrete-time finite-state Markov chains [5,6]. Specifically, an influence model tracks the discrete-valued statuses of network components or sites or nodes, which evolve in discrete time through probabilistic interactions with their graphical neighbors. Formally, let us consider an influence model with $n$ nodes (labeled by $1, \ldots, n$), where each node’s status can be one of $m_i$ values at each time $k = 0, 1, 2, \ldots$. The status of each node evolves stochastically, based on following two-stage update rule (see [6]).

- At each time-step $k$, each node $i$ independently picks a node $j$ (possibly including itself) with some probability $d_{ij}$. The selected node $j$ is referred to as the determining node for node $i$.
- The next status of node $i$ (i.e., status at time $k + 1$) is generated independently according to a probability mass function (PMF), which is specified by the current status of the determining site $j$. Thus, the next status of node $i$ is generated probabilistically based on its current status of determining site $j$.

The influence model has found application both in modeling stochastic propagations in networks and as an algorithmic tool, including in social-network modeling [8], sensor fusion [9], graph partitioning [10] and weather modeling for air traffic management [2]. The applicability of the models stems from its ability to capture a range of stochastic interactions, while also maintaining tractability.

In the following subsections we introduce some notations and some of the results which were developed in [11,5] and [6]. These results show that first (and higher) order statistics of the statuses of nodes in an influence model can be generated from a first-order recursion.

1) Notations, Terminologies and Vector Form Representations: The statistical analysis of the influence model—which we will also leverage in the layered-network modeling framework—requires some further notation. Keeping with the notations introduced earlier we consider an influence model of the $n$ nodes (labeled by $1, \ldots, n$). First, we find it convenient use an indicator-vector notation for each node’s status. Specifically, we define the node-status vector $s_i[k]$, as an $m_i$-element 0–1 indicator vector, where the single unity entry corresponds to the time-$k$ status of the node. Also, we define a $n \times n$ stochastic matrix $D = [d_{ij}]$ as the network influence matrix of the model. It is also natural to define a network graph topology from $D$, that captures the dependencies between the components in the network. Specifically, we define a weighted and directed graph $\Gamma$ with $n$ vertices (labeled by $1, \ldots, n$), which correspond to the influence model nodes. The graph $\Gamma$ has a directed edge from vertex $j$ to vertex $i$ (where $i$ may equal $j$, i.e. self-loops are allowed) if and only if $d_{ij} > 0$, and further the edge is assigned a weight of $d_{ij}$.

The vector $s[k] = \begin{bmatrix} s_1[k] \\ s_2[k] \\ \vdots \\ s_n[k] \end{bmatrix}$ is termed the full status vector of the influence model at time $k$. We also define an $m_i$-component status probability vector $p_i[k]$ for each node $i$ that captures the probabilities that the node $i$ is in each status at time $k$, i.e.

\[
p_i[k] = \begin{bmatrix} P(s_i[k] = [1 \ 0 \ldots 0]) \\ P(s_i[k] = [0 \ 1 \ldots 0]) \\ \vdots \\ P(s_i[k] = [0 \ 0 \ldots 1]) \end{bmatrix}.
\]
The full status probability vector is defined as \[ p[k] = [p_1[k] \ p_2^T[k] \ldots p_n^T[k]]. \]

When a node \( j \) is selected as the determining node for node \( i \) at time \( k \), the next status probability vector \( p_i[k] \) (conditioned on the current state and the determining-node selection) is given by \[ p_i^T[k+1] = s_i^2[k] A_{ji}, \] where \( A_{ji} \) is a fixed row stochastic \( m_j \times m_i \) matrix. We refer to matrices \( A_{ji} \) as the local evolution matrices. The matrix \( A_{ji} \) captures the dependence of node \( i \)'s next status on node \( j \)'s current status, when node \( i \) selects node \( j \) as its determining node. In this notation, we note that the next-status probabilities \( p_i[k+1] \) are computed from \( s[k] \) as described above, and the time-

\( k+1 \) states are generated probabilistically according to the vectors \( p_i[k+1] \).

2) Tractabilities of an Influence Model: Let us present some statistical analyses of the influence model (developed in [5], [6], [11]), with the intention of the highlighting the rich tractabilities of the model. As a preface, we note that the influence model captures a quite high-dimensional Markov-chain dynamics: each site takes on \( m_i \) statuses, so the network as a whole has \( \prod_i m_i \) configurations that evolve in a Markov fashion. For even a moderate-sized network, tracking configuration probabilities is infeasible. Interestingly, however, partial statistics (e.g., individual nodes’ status probabilities, or joint-status probabilities of small groups of nodes), can be found with much less effort. Here, we present the recursion for computing individual sites’ status probabilities (equivalently, the expected values of the nodes’ status vectors). We also point to higher-dimensional recursions for finding joint-status probabilities (equivalently, higher moments of the status vector).

Here is the analysis of the individual sites’ status probabilities. From the update rules for influence model, we see that the conditional status probabilities at the next time step can be generated from the status vector at the current time step. Specifically,

\[ p[k+1] = H s[k] \]

where

\[ H \triangleq \begin{pmatrix} d_{11} A_{11} & \ldots & d_{1n} A_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} A_{n1} & \ldots & d_{nn} A_{nn} \end{pmatrix} = (D^T \otimes \{A_{ij}\})^T \]

is called the full-influence matrix. We note here that the full-influence matrix \( H \) is computed according to a generalization of the Kronecker product, as expressed above. Noting that the status-probability vector is the expectation of the status vector at the corresponding time, the conditional expectation of the time-

\( k+1 \) status vector \( s[k+1] \) given the time-

\( k \) status vector \( s[k] \) can be found:

\[ E[s[k+1] | s[k]] = H s[k] \]

This conditional expectation immediately yields the following recursion for the expected status vector (equivalently, the full probability vector), through the law of iterated expectations:

\[ E(s[k+1]) = H E(s[k]) \]

Modal analysis of the matrix \( H \) has also been developed in [5]. These modal analyses directly characterize the asymptotics of the first moment of the status vector, and also inform understanding of the asymptotic behavior of the influence model as a whole. From the definition of the matrix \( H \), the modal analysis can be connected to the graph structure of the influence model.

Relatively low complexity recursions for second and higher moments of the state process \( s[k] \), or equivalently for joint status probabilities, have been developed in [5] and in [11]. In interest of space we only present a result that shows that second and first order statistics of the influence model can be simultaneously generated through a first-order recursion similar to Equation 5 (see [11] for a detailed development). We use this result later in our analysis of the layered network model.

To develop the recursion, we define a vector \( s[2](k) \) that contains both pair-wise Kronecker products of status vectors of two distinct nodes and individual nodes’ statuses. Specifically, we define

\[ s[2](k) = \begin{bmatrix} s[2](k) \\ s[k] \end{bmatrix}, \]

where \( s[2](k) \) contains Kronecker products of the form \( s_i[k] \otimes s_j[k] \) where \( i = 1, \ldots, n \), \( j = 1, \ldots, n \) and \( i < j \). In particular, \( s[2](k) \) is given by

\[ s[2](k) = \begin{bmatrix} s_1[k] \otimes s_2[k] \\ \vdots \\ s_1[k] \otimes s_n[k] \\ s_2[k] \otimes s_3[k] \\ \vdots \\ s_{n-1}[k] \otimes s_n[k] \end{bmatrix} \]

Please note that \( s[2](k) \) can be obtained from the block Kronecker \( s[k] \otimes s[k] \) by deleting certain components, and is a sparser representation of the same pairwise data. The following recursion for the expectation of \( s[2](k) \) permits us to track individual nodes’ status probabilities and pairwise joint probabilities:

\[ E(s[2](k+1)) = \tilde{H}(2) E(s[2](k)). \]

The thesis [11] provides explicit expressions the matrix \( \tilde{H}(2) \), in terms of the network influence matrix \( D \) and the set of the local evolution matrices \( \{A_{ij}\} \) of the model. Modal analysis of the matrix \( \tilde{H}(2) \) informs asymptotic and settling-time analysis of the model, and has been discussed in [11]. The reduced form representation described above facilitates this modal analysis.

Through the remainder on the paper we use the notation, \( IM(D_{IM}, A_{IM}, s[0]) \) to represent an influence model with \( D_{IM} \) as the network influence matrix, \( A_{IM} \) as the family of local evolution matrices \( \{A_{ij}\} \) and \( s[0] \) as the status vector at initial time \( k = 0 \). We stress that these three parameters completely define an influence model.

3) Homogeneous Influence Model: The homogeneous influence model is a special case of the influence model, in which the local evolution matrices are uniform throughout the network. Specifically, each node has the same number of
possible statuses, \((m_1 = m)\) and all the local evolution matrices are identical \((A_{ij} = A)\). The homogeneous case permits an especially simple asymptotic analysis. Specifically, for the homogeneous influence model, the full influence matrix is a pure Kronecker product: \(H = (D^T \otimes A)^T\). Further \(H\) has a unique strictly-dominant unity eigenvalue when the network topology is irreducible and \(A\) (which is a stochastic matrix) is ergodic. For such a case, the status moments converges to some finite values that are independent of the initial status. The references [5], [6], [11] give much further analysis of the homogeneous influence model.

III. Model Proposal

In this section, we formulate the layered network model. The model consists of two distinct pieces: 1) a network model with \(n_u\) nodes (labeled \(1, \ldots, n_u\)) that captures the evolution of a spatially and temporally correlated discrete-valued environmental-uncertainty process; 2) a second network model consisting of \(n_i\) nodes (labeled \(1, \ldots, n_i\)) that captures a state evolution of interest for an infrastructure. To avoid confusion, we refer to nodes in the environmental-uncertainty network model as environmental nodes, and those corresponding the infrastructure network dynamics as infrastructure nodes.

The environmental uncertainties evolve in discrete time, and are independent of the infrastructure network dynamics. Specifically, at every time step, the \(i^{th}\) environmental node may have one of \(m_i\) possible statuses (which evolve through interactions with other environmental nodes). To represent the status of the \(i^{th}\) environmental node at the \(k^{th}\) time-step, we use an \(m_i\) length indicator vector, \(y_i[k]\). We represent the statuses of the environmental uncertainties across the entire network as \(y[k] = \begin{bmatrix} y_1[k] \\ \vdots \\ y_{n_u}[k] \end{bmatrix}\). Please note that for physical uncertainty processes, such as weather, each of the \(n_u\) environmental nodes may correspond to a geographical location. However, such a correspondence is not a necessity, and the environmental nodes can represent abstract or tangible quantities depending on the application.

The infrastructure dynamics also evolve in discrete-time. The \(i^{th}\) infrastructure node is assumed to be tracking/modeling the evolution of some continuous valued physical quantities which we refer to as the state variables for the node. We allow flexibility in the number of the state variables tracked by an infrastructure node. The state variables are represented as a vector denoted as \(x_i[k]\), which we refer to as the state of \(i^{th}\) infrastructure node. The vector \(x[i][k] = \begin{bmatrix} x_1[k] \\ \vdots \\ x_{n_i}[k] \end{bmatrix}\) captures the state of the entire infrastructure dynamics. We will model the state variable of an infrastructure node as evolving through linear interactions with neighboring nodes as well as linear driving signals from the environmental uncertainty network.

The above is an overview of the two pieces of our layered network model and their interactions. Let us now describe the specifics of the model.

We model the environmental uncertainty process \(\{y[k]\}\) as being generated by an influence model \(IM(D_{IM}, A_{IM}, y[0])\). That is, the vector of statuses, \(y[k], k = 1, 2, 3, \ldots\) are generated by influence model. We assume that \(D_{IM}\) and \(A_{IM}\) are dimensionally consistent with the number of environmental nodes and the number of possible statuses for each environmental node (see Section II for details). We also assume that the influence model for the environmental uncertainties is known. For a description of how an influence model for environmental uncertainties can be parameterized from forecast snapshots, we ask our readers to see [2]. We note that the matrix \(D_{IM}\) defines the underlying topology of the network model for environmental uncertainty evolution.

Meanwhile, the infrastructure node’s state is modeled as evolving through linear interaction with other infrastructure nodes, and is also driven by a linear projection of the (discrete-valued) environmental uncertainties. That is, the state \(x_i[k]\) evolves according to the following equation.

\[
x_i[k+1] = \sum_{j=1}^{n_i} \tilde{A}_{ij}[k]x_j[k] + B_i[k]u_i[k]
\]

where, \(u_i[k] = G_i[k]y[k]\)

We assume that \(\tilde{A}_{ij}, B_i, G_i\) are dimensionally consistent with \(x_i[k], x_j[k]\) and \(y[k]\). We refer to the block matrix \(A[k] = [\tilde{A}_{ij}]\) as the system matrix of infrastructure network. We stress that the two network layers follow the same clock, and are initialized concurrently. We denote the initial time as \(k = 0\).

For simulation and analysis purposes, it is helpful to present the layered network model in full as a discrete time linear system that is driven by the state process of an influence model:

\[
x[k+1] = A[k]x[k] + B[k]u[k], \quad u[k] = G[k]y[k], \quad IM(D_{IM}, A_{IM}, y[0]) \rightarrow \{y[k]\}
\]

where,

\[
x[k] = [x_1^T[k] \quad x_2^T[k] \quad \ldots \quad x_n^T[k]]^T, \quad A[k] = [\tilde{A}_{ij}], \quad B[k] = \begin{bmatrix} B_1[k] \\ \vdots \\ B_n[k] \end{bmatrix}, \quad G[k] = [G_1^T[k] \quad G_2^T[k] \quad \ldots \quad G_n^T[k]]^T.
\]

It is also natural to define a graph for the layered network model, that captures interactions among nodes in each layer and between layers. For the basic statistical analyses of the layered network model pursued here, detailed notation regarding the graph topology is not needed and so we omit a formal definition here.

IV. Some Analysis of the Model

In this section we present some preliminary statistical analyses of the layered network model. Specifically, we demonstrate that the local first and second order statistics of
the infrastructure network dynamics and the environmental-uncertainty statuses (Equation 9), can be found using a relatively low-dimension recursion. We formalize these results in theorems 1 and 2. As a corollary we specialize the results to the case where the infrastructure network is time invariant and the influence model driving it is homogeneous. We also discuss a notion of stability for the model and present a sufficient condition for it.

As a first step, we define an augmented network state vector $\tilde{x}[k] = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$, to facilitate the statistical analysis of the layered network model. Using Equation 9 for $x[k]$ and Equation 5 for $y[k]$, we obtain:

$$E(\tilde{x}[k+1]|\tilde{x}[k]) = \hat{A}_1[k]\tilde{x}[k]$$

where, $\hat{A}_1[k] = \begin{bmatrix} A[k] & B[k]G[k] \\ 0 & H \end{bmatrix}$. Here [0] represents a block zero matrix of appropriate dimension. Applying the expectation operator on both sides of Equation 10, we arrive at the following result.

**Theorem 1:** Consider the layered model presented in Equation 9. The first moment (mean) of the augmented vector $\tilde{x}[k] = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$ is governed by

$$E(\tilde{x}[k+1]) = \hat{A}_1[k]E(\tilde{x}[k])$$

Thus given the expected values of $\tilde{x}[0]$, the mean of the infrastructure and environmental-uncertainty dynamics can be simultaneously computed for all future times, using a simple first-order recursion. We now proceed to develop results for the second order statistics of the model again using the augmented vector $\tilde{x}[k]$. Since $\tilde{x}[k]$ is naturally partitioned into an infrastructure state vector and an environmental-uncertainty state vector, $\tilde{y}[k] = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$, we find the block Kronecker product (see Section II-A for a brief review) convenient in representing the second order statistics. Our focus is in charactering the expectation of

$$(\tilde{x}[k+1] \otimes \tilde{x}[k+1]) = \begin{bmatrix} x[k+1] \otimes x[k+1] \\ x[k+1] \otimes y[k+1] \\ y[k+1] \otimes x[k+1] \\ y[k+1] \otimes y[k+1] \end{bmatrix},$$

which contains second order statistics of state and status variables. However, we notice that $(\tilde{x}[k+1] \otimes \tilde{x}[k+1])$ contains redundant terms, which are worthwhile to remove so as to reduce model analysis complexity and simply the spectral analysis of the model. Further, the second order statistics of the $y[k]$ depend on its first order statistics (Equation 7), and hence a representation that automatically tracks the first-order statistics of $y[k]$ is convenient. Therefore we define a reduced second-order state vector $\tilde{x}_2[k]$ that removes redundant terms and includes the vector $y_2[k]$, i.e.

$$\tilde{x}_2[k] = \begin{bmatrix} x_2[k] \\ x_2[k] \otimes y_2[k] \\ y_2[k] = y_2[k] \end{bmatrix}$$

Here, $x_2[k]$ is a vector of components of $x[k] \otimes x[k]$ without second occurrence of redundant entries. The following theorem presents the single step recursion for the expected value of the reduced second order state vector.

**Theorem 2:** Consider the layered model presented in Equation 9. The expected value of the reduced second order state vector is governed by the following single step recursion.

$$E(\tilde{x}_2[k+1]) = \hat{A}_2[k]E(\tilde{x}_2[k+1])$$

where, $\hat{A}_2$ has the following block structure

$$\hat{A}_2 = \begin{bmatrix} P_1(\hat{A} \otimes A)Q_1(\hat{A} \otimes B)G(\hat{A} \otimes A) & P_2 & P_3(\hat{A} \otimes B)G(\hat{A} \otimes A)Q_4 & 0 \\ 0 & \hat{A} \otimes H & 0 & 0 \\ 0 & 0 & \hat{A} \otimes H & P_4(\hat{A} \otimes B)G(\hat{A} \otimes A)Q_4 \\ 0 & 0 & 0 & \hat{H} \end{bmatrix}$$

and where, $P_1$ and $Q_1$ are certain $I \times I$ rectangular matrices, whose each row is an indicator vector. Please note that all system matrices i.e. $A$, $B$ and $G$, are time-varying. The time argument $k$ has been dropped for notional convenience. Further, please note that the block matrix $\hat{H}_2$ is dimensionally consistent with the block representation of $\hat{A}_2$.

The result can be proved by considering $E(\tilde{x}_2[k+1]|\tilde{x}_2[k])$. Each block in this expression can be written as a linear function of $\tilde{x}_2[k]$, either by invoking the statistical analysis of the influence model or from expressions for $x[k+1]$ and $x[k+1] \otimes x[k+1]$ (together with the expansion/contraction between the reduced and full second order vectors). The analysis becomes somewhat cumbersome, so we exclude the details in the interest of space. We note that the matrices $P_1$ and $Q_1$ serve to expand/contract/re-order the blocks in $\tilde{x}_2[k]$ to convenient forms for analysis, i.e. these introduce or exclude redundancies as needed; thus they have the row-wise indicator structure, as specified in the theorem.

A special case of the layered network model is that the parameters $A[k]$, $B[k]$ and $G[k]$ are time-invariant and the influence model is homogeneous. This case is of particular interest to us as various layered network architecture can be represented with this model. Time-invariant infrastructure networks are typical in engineered networks such as air traffic network [4]. Homogeneous influence models have been used in automated chip design [12] and resource allocation over networks[13]. In addition, this case is mathematically speciﬁcally tractable; various qualitative insights about the model can be obtained easily from spectral analysis of the moment recursions matrices in this case. We present the moment recursions matrices for this case explicitly in the next corollary.

**Corollary 1:** If the systems described in (9) are time-invariant and the influence model driving the network linear systems is homogeneous, then,

$$\hat{A}_1 = \begin{bmatrix} A & BG \\ 0 & D_{IM} \otimes \hat{A}_{IM} \end{bmatrix}$$

$$\hat{A}_2 = \begin{bmatrix} P_1(\hat{A} \otimes A)Q_1(\hat{A} \otimes B)G(\hat{A} \otimes A) & P_2 & P_3(\hat{A} \otimes B)G(\hat{A} \otimes A)Q_4 & 0 \\ 0 & \hat{A} \otimes H & 0 & 0 \\ 0 & 0 & \hat{A} \otimes H & P_4(\hat{A} \otimes B)G(\hat{A} \otimes A)Q_4 \\ 0 & 0 & 0 & \hat{H} \end{bmatrix}$$

Please see [11] for construction of the block $\hat{H}_2$ for an homogeneous influence model. For the time-invariant homogeneous case, we refer to $\hat{A}_1$ as the first moment recursion matrix and $\hat{A}_2$ as the second moment recursion matrix.

In concluding the moment analysis, let us stress that the developed recursion permit relatively low-complexity analysis of a hybrid-highly stochastic network dynamics. In particular, the influence model constitutes a very high dimensional Markov chain (with $\prod m_i$ states), and further the layered network model captures a continuous valued
process driven by such discrete-valued stochastic evolutions. The special structure in the model permits us to track partial statistics of the dynamic using lower-order recursions. Let us now discuss some qualitative features of the model in next sub-section, IV-A.

A. Initial Stability Analysis

So far, we have presented recursions that yield first- and second-order statistics for the layered network model. Let us now use these recursions to characterize asymptotic properties of dynamics. The layered network model is general enough to permit a wide variety of asymptotic dynamics. In this initial exposition, we consider the circumstance that the infrastructure dynamics is being driven by a persistent environmental stimulation. We are concerned with whether or not the first and second moments of the state/statuses reach finite asymptotes (that are independent of the initial condition). We refer to this particular notion as moment convergence.

Let us develop conditions for moment convergence when the layered network model is time-invariant and has a homogeneous influence model. Sufficient conditions to ensure moment convergence is presented in the next theorem.

**Theorem 3:** Consider the layered network model presented in Equation 9. Assume that the model is time-invariant. Also assume that the influence model $IM(D_{IM}, A_{IM}, s[0])$ is homogeneous. Further, assume that $A_{IM}$ and $D_{IM}$ are ergodic matrices. If all the eigenvalues of the block matrix $A$ are strictly within the unit circle, the first and second moments of the layered network model are convergent, i.e., after enough time has passed, the statistics converge to steady state values that do not depend on the initial condition.

**Proof:** The basic philosophy of our proof is to show that the matrices $A_1$ and $A_2$ meet the well-known conditions on the eigenvalues of a system-matrix of the unforced, discrete, linear time-invariant system that ensure BIBO stability. We omit the proof in interest of space and refer our readers to the extended version of this paper [14].

The above result does not provide information about steady state or settling time of the recursion, only that an asymptote is reached. To be able to find the steady states, one needs to find the eigenvectors of the recursion matrices. Due to the complex structure of the matrices (especially $A_2$), a structural characterization of the eigenvectors is a non-trivial, especially since eigenvector analysis of Kronecker products of defective matrices is difficult, see [15], [16].

V. Conclusions

A two-layered network model has been developed, that is promising for representing linear infrastructure-network dynamics that are modulated by complex discrete-valued environmental-uncertainty processes. The developed hybrid model has a moment-closure property, that permits relatively low-complexity statistical analysis of the meshed dynamics. Here, we have presented analyses of the first two state moments, and used these analyses to gain some insight into the layered model’s asymptotics in the case where the environmental impacts are persistent. It is worth noting that the Kronecker product formalism that we have used for moment analysis also naturally permits characterization of higher moments. We also stress that the moment analyses can permit much further analysis of asymptotics, see for instance the methods in [11]. We defer such further analysis to future work.

Finally, we ask our readers to refer to the extended document [17], which includes a detailed example to illustrate the layered-network-model dynamics.

**REFERENCES**