An Adjacency Matrix Approach to Delay Analysis in Temporal Networks

John M. Shea
Wireless Information Networking Group
University of Florida, Gainesville FL
jshea@ece.ufl.edu

Joseph P. Macker
Information Technology Division
Naval Research Laboratory, Washington DC
joseph.macker@nrl.navy.mil

***Abstract*—Wireless communications networks are often modeled as graphs in which the vertices represent wireless devices and the edges represent the communication links between them. However, graphs fail to capture the time-varying nature of wireless networks. Temporal networks are graphs in which the sets of nodes or edges are time-varying. We consider the most common case, in which the set of nodes is fixed but the presence of edges changes over time. Most previous work on analyzing temporal networks has focused on summary measures that combine the contributions of different paths by using different weights for paths with different delays. Such summary measures are efficient to compute but may lose valuable information about the temporal behavior of the network. We propose techniques that characterize the delays of all paths between nodes in temporal networks. We then apply these techniques to identify dominant patterns in the temporal paths connecting nodes. Example temporal networks are used to illustrate these phenomena, and we consider implications to wireless networks.**

I. Introduction

Graphs are commonly used to model a wide variety of systems in which a group of entities are represented by the vertices and the relations or connections between them are represented by the edges. For instance, in wireless networks, radios are often represented by vertices, and the presence of an edge between two vertices indicates that the associated radios share a communication link. However, in many real systems, the entities and/or their relationships and connections change with time. The notion of graphs can be extended to capture these time dependencies, in which case they are referred to as *temporal networks* [1, 2]. The most common class of temporal networks are those with a fixed set of vertices, but a time-varying set of edges. For instance, in wireless networks, the edges may represent communication links that vary over time because of motion, fading, or jamming.

The time-varying nature of temporal networks make analyzing them more complicated than graphs. Moreover, tools from graph theory cannot directly apply to temporal networks. For instance, the paths between nodes in a graph are transitive: if node A can reach node B and node B can reach node C, then node C can be reached from node A. The same is not necessarily true in a temporal network [1, 2]. The maximum set of disconnected paths between vertices in a graph can be found in polynomial time, but to even determine whether there are two disjoint time-respecting paths in a temporal network is NP-complete [3]. Most authors have focused on extending summary measures used to analyze importance of nodes to graph connectivity from graphs to temporal networks. For instance, in [4], the authors introduce temporal closeness centrality and temporal betweenness centrality. A survey of these and other metrics is in [3]. Methods for calculating temporal distances are described in [5].

This research was funded in part by the Naval Research Laboratory's Characterization and Performance Prediction in Ad hoc Networks (CAPPAN) project and by the National Science Foundation under grant number 1642973.

Fig. 1. Four node linear network.

The goal of the work presented in this paper is to move beyond summary measures of connectivity and develop new techniques to compute the number of end-to-end paths with a given delay between any two vertices. Our approach is to use a form that is analogous to the usual product of adjacency matrices for computing the number of paths between two vertices. We then use time-domain decompositions and visualizations to extract meaning from these measures.

II. Temporal Networks and Differential Delay

Consider temporal networks of the form $\mathcal{G} = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V}$ is a set of vertices that does not change over time and $\mathcal{E}(t)$ is a time-varying set of edges such that $\mathcal{E}(t) \in \mathcal{V} \times \mathcal{V}$. Let $\overline{\mathcal{G}} = (\mathcal{V}, \cup_t \mathcal{E}(t))$ denote the non-temporal network that results from superimposing the sets of edges that exist at all times. Throughout this paper, we assume that $\mathcal{V}$ represents a set of communicators, the edges $\mathcal{E}(t)$ represent communication links among them, and the communicators wish to send information over the edges.

Consider first a *linear temporal network*, which is defined as a temporal network $\mathcal{G}$ for which $\overline{\mathcal{G}}$ is a linear network. In such a network, the nodes can be drawn on a horizontal line such that the nodes only share edges with the nodes that are immediately to their left or right. In addition, we assume that $\mathcal{G}$ is simple for every t. That is there are no self-loops and the edges are undirected. We also assume that the edges are unweighted and that $\overline{\mathcal{G}}$ is connected.

An example of such a network with four nodes is shown in Fig. 1. In a drawing of a temporal network, we can label

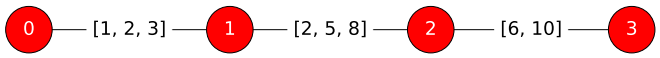

Fig. 2. Four node temporal network.

Input Link Time	Output Link Time	Delay
1	2	1
1	5	4
1	8	7
2	5	3
2	8	6
3	5	2
3	8	5

TABLE I
POSSIBLE COMMUNICATION FROM NODE 0 TO NODE 2 AND RESULTING DELAYS FOR EXAMPLE FOUR-NODE NETWORK.

each edge with the times at which that edge is present in the network. Consider the following scenario for the edges:

- $0 \leftrightarrow 1$ is available during times 1, 2, and 3;
- $1 \leftrightarrow 2$ is available during times 2, 5, and 8; and
- $2 \leftrightarrow 3$ is available during times 6 and 10.

Then the temporal network can be drawn as shown in Fig. 2.

For a temporal network, we require that walks be time-respecting. That is, the walk must specify not only the set of edges connecting the nodes, but also an ordered set of times at which those edges exist. Mathematically, we can express a walk $\mathcal{W}$ as

$$\mathcal{W} = \bigcup_{n\in\{1,2,\ldots N\}} j_{n-1} \overset{t_n}{\rightarrow} j_n,$$

where $j_n \in \mathcal{V}$ for all $n \in \{0, 1, \ldots, N\}$, $(j_{n-1}, j_n) \in \mathcal{E}(t_n)$ for all $n \in \{1, 2, \ldots, N\}$, and $t_1 < t_2 < \ldots < t_N$.

We define the *differential delay* across two links as the difference in time between when the information can be transmitted across the first link and when it can be transmitted across the second link. In this paper, we assume that information cannot flow over multiple edges in a single time, and so for a given input link time, the output link can only be used at times that are strictly greater than the input link time. The analytical methods developed in this paper are easily extendable to cases where information can flow over multiple links in a single time. Thus, for links $0 \leftrightarrow 1$ and $1 \leftrightarrow 2$ in the example shown in Fig. 2, the possible communication times using these links and the associated differential link times shown in Table I.

Similarly, we can find the end-to-end differential delay over a path consisting of multiple consecutive edges. Here, the end-to-end differential delay is defined as the difference in the times between when information is first transmitted over the first link in the path to when it is delivered to the destination over the last link in the path. For the links $0 \leftrightarrow 1$, $1 \leftrightarrow 2$, and $2 \rightarrow 3$, a few of the possible paths and the associated end-to-end differential delays are shown in Table II.

Link $0 \rightarrow 1$ Time	Link $1 \rightarrow 2$ Time	Link $2 \rightarrow 3$ Time	Differential Delay
1	2	6	5
1	5	6	5
2	5	6	4
3	5	6	3
1	8	10	9

TABLE II
SEVERAL VALID COMMUNICATION TIMES FROM NODE 0 TO NODE 3 AND RESULTING DELAYS FOR EXAMPLE FOUR-NODE NETWORK.

III. TEMPORAL ADJACENCY MATRICES

As shown in Section II, the differential delays for a temporal network can be found by enumerating all of the temporal walks between two nodes and then calculating the delays. However, it is very challenging to enumerate all of the temporal walks for even small networks, even when the number of time instances is less than 10. Thus, in this paper, we present a mathematical technique to enumerate the end-to-end delays via an approach similar to how the adjacency matrix is used for enumeration of paths in non-temporal graphs.

Consider a temporal graph $\mathcal{G}$ with a fixed set of N vertices/nodes, but where the set of edges changes depending on the time. Without loss of generality, we assume that the vertices are labeled $0, 1, \ldots, N-1$. For our purposes, we assume that the set of edges is defined on a time index set of the form $\{0, 1, \ldots, T-1\}$.

We propose a new *temporal adjacency matrix* (TAM) for $\mathcal{G}$, which is an $N \times N$ matrix in which entry (i, j) contains information about the times in which node i can communicate with node j. We use $\mathcal{A}$ to denote the TAM to distinguish it from the usual adjacency matrix $\mathbf{A}$. We use the notation $\mathcal{A}_{i,j}$ to denote the entry of $\mathcal{A}$ in the ith row and jth column.

A. Temporal Adjacency Matrix Entries

As with the usual adjacency matrix, the diagonal entries of $\mathcal{A}$ are set to the integer 0; $\mathcal{A}_{i,i} = 0$ for $i = 0, 1, \ldots, N-1$. If there is not an edge connecting vertex i to vertex j at any time, then $\mathcal{A}_{i,j} = 0$. Unlike a usual adjacency matrix, if nodes i and j share an edge at any time, the (i, j)th entry of the temporal adjacency matrix is itself a matrix. Such entries are defined in a way that will allow us to determine the path delays between any two nodes through a type of product of temporal adjacency matrices, just as products of the usual (unweighted) adjacency matrix allow us to count the number of paths connecting two vertices. The nonzero entries of a temporal adjacency matrix are in the form of a partitioned matrix,

$$\mathcal{A}_{i,j} = \left[\begin{array}{c} \mathcal{C}_{i,j} \\ \hline \mathcal{D}_{i,j} \end{array}\right].$$

We begin by defining $\mathcal{D}_{i,j}$. Let $\mathbb{T}_{i,j}$ be a row vector containing the time indices of when there is an edge from node i to node j. If

$$\mathbb{T}_{i,j} = \left[\begin{array}{cccc} t_1 & t_2 & \ldots & t_M \end{array}\right],$$

then

$$\mathcal{D}_{i,j} = \left[\begin{array}{cccc} s^{t_1} & s^{t_2} & \ldots & s^{t_M} \\ s^{-t_1} & s^{-t_2} & \ldots & s^{-t_M} \end{array}\right],$$

and $\mathcal{C}_{i,j} = \mathbf{I}_M$, the identity matrix of size $M \times M$. This form will facilitate computation of the possible end-to-end delays, as shown further below.

For matrix E that is an element of a temporal adjacency matrix, we use $\mathcal{C}(E)$ and $\mathcal{D}(E)$ to refer to the component matrices in the form

$$E = \left[\begin{array}{c} \mathcal{C}(E) \\ \hline \mathcal{D}(E) \end{array}\right],$$

where $\mathcal{D}(E)$ is the last two rows of E.

Consider again the temporal network shown in 2. Link $0 \to 1$ is available at times 1,2,3, and so the temporal adjacency matrix entry $\mathcal{A}_{0,1}$ is

$$\mathcal{A}_{0,1} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline s^1 & s^2 & s^3 \\ s^{-1} & s^{-2} & s^{-3} \end{array}\right]$$

Link $1 \to 2$ is up at times 2,5,8, and so $\mathcal{A}_{1,2}$ is

$$\mathcal{A}_{1,2} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline s^2 & s^5 & s^8 \\ s^{-2} & s^{-5} & s^{-8} \end{array}\right]$$

B. Defining Multiplication for Entries of the Temporal Adjacency Matrix

As explained above, the entries of a temporal adjacency matrix are themselves matrices (or else 0). We will need to perform a multiplication-like operator on the entries in computing the delays using the temporal adjacency matrices. Let $\odot$ be the multiplication operator for two entries of the temporal adjacency matrix. We define $\odot$ below.

Consider the product $L \odot R$, where L and R are entries of a temporal adjacency matrix. Later we will generalize this definition to apply to other classes of matrices created from temporal adjacency matrices. If either $L = 0$ or $R = 0$, we define $L \odot R = 0$.

Now consider the case where both $L \neq 0$ and $R \neq 0$. $\mathcal{C}(L)$ is multiplied by a two-link delay matrix created from $\mathcal{D}(L)$ and $\mathcal{D}(R)$. For the case where L and R are elements of temporal adjacency matrices, $\mathcal{C}(L)$ is an identity matrix, and we only need concern ourselves with the two-link delay matrix. However, below, we show a general technique that can still be applied when L is not a temporal adjacency matrix, which we will utilize in Section III-D.

We begin to investigate the necessary computation by utilizing the example from Fig. 2. For the pair of links $0 \to 1$ and $1 \to 2$, we first compute the outer product of $\mathcal{D}(L)_2$ and $\mathcal{D}(R)_1$, which we denote by T.

$$\begin{aligned} T &= \left[\begin{array}{c} s^{-1} \\ s^{-2} \\ s^{-3} \end{array}\right] \cdot \left[s^2 \quad s^5 \quad s^8\right] = \mathcal{D}(L)_2^T \mathcal{D}(R)_1 \\ &= \left[\begin{array}{ccc} s & s^4 & s^7 \\ 1 & s^3 & s^6 \\ s^{-1} & s^2 & s^5 \end{array}\right], \end{aligned}$$

where $\mathcal{D}(X)_j$ denotes the jth row of $\mathcal{D}(X)$.

Note that for the (i,j)th entry, the power of s represents the difference in times between the ith time the input path was available and the jth time the output path was available. These are the possible differential delays for information flowing over the two links except that negative delays are not feasible, because the information would use the second link at a time before it traveled over the first link. To preserve only the possible two-link delays, we must eliminate the entries with nonpositive exponents by replacing those entries with 0.

Define the function $S^+(s^m)$ by

$$S^+(s^m) = \begin{cases} s^m, & m > 0 \\ 0, & m <= 0 \end{cases}.$$

If T is a matrix with entries $t_{i,j}$ of the form $s^{m_{i,j}}$, then define $S^+(T)$ as the matrix with entries $S^+(t_{i,j})$. Thus, for the example,

$$\tilde{T} = S^+(T) = \left[\begin{array}{ccc} s & s^4 & s^7 \\ 0 & s^3 & s^6 \\ 0 & s^2 & s^5 \end{array}\right]$$

Finally, we let

$$\mathcal{C}(L \odot R) = \mathcal{C}(L) \cdot \tilde{T},$$

$$\mathcal{D}(L \odot R) = \mathcal{D}(R),$$

and the final multiplication operation for entries in a temporal adjacency matrix is:

$$\begin{aligned} L \odot R &= \left[\begin{array}{c} \mathcal{C}(L) \cdot S^+(\mathcal{D}(L)_2^T \mathcal{D}(R)_1) \\ \hline \mathcal{D}(R) \end{array}\right] \\ &= \left[\begin{array}{ccc} s & s^4 & s^7 \\ 0 & s^3 & s^6 \\ 0 & s^2 & s^5 \\ \hline s^2 & s^5 & s^8 \\ s^{-2} & s^{-5} & s^{-8} \end{array}\right] \end{aligned}$$

By inspecting this matrix, we can determine all the possible delays and the output link times from which they are derived. In particular, for the output link at time 2, look at column 1; for the output link at time 5, look at column 2; for the output link at time 8, look at column 3. Then we see that for packets coming from the output link at time 2, the only possible delay is 1 (there is only an s^1 term). This is because the only time the packets could have propagated across the first link is at time 1, resulting in a path delay of 1. For the output link at time 5, the possible delays are 2, 3, or 4. The packets could have come from the first link at times 1, 2, 3. Similarly, for the output link at time 8, the possible delays are 5, 6, or 7.

C. Cumulative Differential Path Delays

The exponents of the entries in $\mathcal{C}(L \odot R)$ are all possible nonzero delays for the path with links that have temporal adjacency matrix entries L and R. Moreover, all path delays are associated with the time that the last link is used, which is the power of s in the first row of $\mathcal{D}(L \odot R)$. Thus, we call $L \odot R$ a cumulative differential path delay (CDPD) matrix.

The power of the CDPD matrix is that it can then be used to compute the delay on an additional link by just multiplying (using $\odot$) with the entry of the temporal adjacency matrix for that link. Consider a sequence of operations $(\mathcal{A}_{i,j} \odot \mathcal{A}_{j,k}) \odot \mathcal{A}_{k,\ell}$. For concreteness, let us use the example network above and consider $(\mathcal{A}_{0,1} \odot \mathcal{A}_{1,2}) \odot \mathcal{A}_{2,3}$. Let $Q = \mathcal{A}_{0,1} \odot \mathcal{A}_{1,2}$. Since $\mathcal{D}(Q) = \mathcal{D}(A_{1,2})$, the same arguments as before show that $V = S^+\left(\mathcal{D}(Q)_2^T \mathcal{D}(\mathcal{A}_{2,3})_1\right)$ is a matrix of the delays between the edge $1 \leftrightarrow 2$ and the edge $2 \leftrightarrow 3$. Mathematically, we have

$$\begin{aligned} V &= S^+(\mathcal{D}(Q)_2^T \cdot \mathcal{D}(A_{2,3})_1) = S^+(\mathcal{A}_{1,2})_2^T \cdot \mathcal{D}(A_{2,3})_1) \\ &= S^+\left(\begin{bmatrix} s^{-2} \\ s^{-5} \\ s^{-8} \end{bmatrix} \cdot \begin{bmatrix} s^6 & s^{10} \end{bmatrix}\right) \\ &= S^+\left(\begin{bmatrix} s^4 & s^8 \\ s^1 & s^5 \\ s^{-2} & s^2 \end{bmatrix}\right) = \begin{bmatrix} s^4 & s^8 \\ s^1 & s^5 \\ 0 & s^2 \end{bmatrix} \end{aligned}$$

Note that each row of this matrix represents the delays that are possible for one of the possible up times of the link $1 \leftrightarrow 2$. In particular, the first row represents the additional delays when the information flows over $1 \leftrightarrow 2$ at time 2, the second row is for time 5, and the third row is for time 8. Now, $\mathcal{C}(Q \odot \mathcal{A}_{1,2}) = \mathcal{C}(Q) \cdot V$

The following lemma summarizes properties of the CDPD matrix for linear temporal networks.

Lemma 1. *Let $\mathcal{G}$ be a linear temporal network, and let $\mathcal{H}$ denote the CDPD matrix given by*

$$\mathcal{H} = \mathcal{A}_{j_0,j_1} \odot \mathcal{A}_{j_1,j_2} \odot \ldots \odot \mathcal{A}_{j_{N-1},j_N},$$

for a total of N products.

1. *The (j,k)th entry of $\mathcal{C}(\mathcal{H})$ can be written in the form $N_d s^d$.*
2. *The nonzero entries of the kth column of $\mathcal{C}(\mathcal{H})$ are unique.*
3. *If the $(1,j)$th entry of $\mathcal{D}(\mathcal{A}_{j_0,j_1})$ is s^α and the $(1,k)$th entry of $\mathcal{D}(\mathcal{H})$ is s^β, and the (j,k)th $\mathcal{C}(\mathcal{H})$ is $N_d s^d$, then N_i is the number of temporal paths from node j_0 to node j_N with differential delay $d = \beta - \alpha$ and for which the path $j_0 \to j_1$ is taken at time α and the path $j_{N-1} \to j_N$ is taken at time β.*

Proof: By induction on the number of edges traversed, N.

Basis Let N=2. Since the network is linear, all valid paths are of the form

$$\mathcal{P} = \left(i \xrightarrow{t_1} j\right) \cup \left(j \xrightarrow{t_2} k\right),$$

where $t_2 > t_1$.

Then

$$\begin{aligned} \mathcal{H} = \mathcal{A}_{i,j} \odot \mathcal{A}_{j,k} &= \left[\frac{\mathcal{C}(\mathcal{A}_{i,j}) \cdot S^+\left[\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1\right]}{\mathcal{D}(\mathcal{A}_{j,k})}\right] \\ &= \left[\frac{S^+\left[\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1\right]}{\mathcal{D}(\mathcal{A}_{j,k})}\right], \end{aligned}$$

where the last step follows from $\mathcal{C}(\mathcal{A}_{i,j}) = \mathbf{I}$

For convenience of exposition, let

$$\mathbf{U} = (\mathcal{A}_{i,j})_2 = \begin{bmatrix} s^{-u_1} & s^{-u_2} & \ldots & s^{-u_K}, \end{bmatrix}$$

where $\{u_1, u_2, \ldots u_K\}$ are the times that the link from i to j is available. Similarly, let

$$\mathbf{V} = (\mathcal{A}_{i,j})_1 = \begin{bmatrix} s^{v_1} & s^{v_2} & \ldots & s^{v_L} \end{bmatrix},$$

where $\{v_1, v_2, \ldots, v_L\}$ are the times that the link from j to k is available.

Then the (m,n)th entry of $\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1$ is $s^{-u_m} s^{v_n} = s^{v_n - u_m}$, and the (m,n)th entry of $S^+\left[\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1\right]$ is $s^{v_n - u_m}$ if $v_n > u_m$ and 0 otherwise.

The condition that $v_n > u_m$ is the causality condition that the link $i \to j$ is taken before the link $j \to k$. Moreover, when $v_n > u_m$, $v_n - u_m$ is the differential delay across the links.

Thus, for every time u_m that the link $i \to j$ is available and every time v_n that the link $j \to k$ is available, the (m,n)th entry of $\mathbf{H}$ is $s^{v_n - u_m}$ if $v_n > u_m$ and 0 otherwise. Note that for a given n, the possible values of $v_n - u_m$ are unique, since $\{u_1, u_2, \ldots u_K\}$ are unique. Thus, any nonzero term in the nth column of $S^+\left[\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1\right]$ is unique and is of the form s^d for some d.

Note that the $(1,n)$th entry of $\mathcal{D}(\mathcal{H})$ is s^{v_n}, the $(1,m)$th entry of $\mathcal{D}(\mathcal{A}_{i,j})$ is s^{u_m}, and the (m,n)th entry of $\mathcal{C}(\mathcal{H})$ is $s^{v_n - u_m} = s^d$ if $v_n > u_m$ and 0 otherwise. For a given u_m and v_n, there is either one temporal path connecting the two states or none (if the causality condition is violated). Thus, if the (m,n)th entry is written as $N_d s^d$, $N_d \in \{0,1\}$ is the number of temporal paths connecting state i to state k, where the edge $i \to j$ is taken at time $d - v_n$ and the edge $j \to k$ is taken at time v_n.

Inductive step Now, consider a path of N edges, and assume the lemma holds up to time $N-1$.

Let

$$\tilde{\mathcal{H}} = \mathcal{A}_{j_0,j_1} \odot \mathcal{A}_{j_1,j_2} \odot \ldots \odot \mathcal{A}_{j_{N-2},j_{N-1}}$$

and

$$\mathcal{H} = \tilde{\mathcal{H}} \odot \mathcal{A}_{j_{N-1},j_N}.$$

Similar to before, let

$$\mathbf{U} = (\mathcal{A}_{j_0,j_1})_2 = \begin{bmatrix} s^{-u_1} & s^{-u_2} & \ldots & s^{-u_K}, \end{bmatrix}$$

where $\{u_1, u_2, \ldots u_K\}$ are the times that the link from j_0 to j_1 is available. Similarly, let

$$\mathbf{V} = (\mathcal{A}_{j_{N-2},j_{N-1}})_1 = \begin{bmatrix} s^{v_1} & s^{v_2} & \ldots & s^{v_L} \end{bmatrix},$$

where $\{v_1, v_2, \ldots, v_L\}$ are the times that the link from j_{N-2} to j_{N-1} is available. Let

$$\mathbf{W} = (\mathcal{A}_{j_{N-1},j_N})_1 = \left[\begin{array}{cccc} s^{w_1} & s^{w_2} & \ldots & s^{w_M} \end{array} \right],$$

where $\{w_1, w_2, \ldots, w_M\}$ are the times that the link from j_{N-1} to j_N is available.

The (m,n)th term of $\tilde{\mathcal{H}}$ can be written as $N_{m,n}s^{d_{m,n}}$, where $d_{m,n} = v_n - u_m$. The (n,q)th term of $S^+\left[\mathcal{D}(\mathcal{A}_{i,j})_2^T \mathcal{D}(\mathcal{A}_{j,k})_1\right]$ is $S^+[s^{w_q - v_n}]$.

Thus, the (m,q)th term of $\mathcal{H}$ is

$$\sum_{n=1}^{L} N_{m,n} s^{v_n - u_m} s^+\left[s^{w_q - v_n}\right] = s^{w_q - u_m} \sum_{n=1}^{L} N_{m,n} 1_{w_q > v_n}. \tag{1}$$

(Property 1 is proved.)

Note that for the qth column of $\mathcal{D}(\mathcal{H})$, the entries are of the form $N_{m,q}s^{w_q - u_m}$, where the $\{u_m\}$ are unique. Thus, the entries in the qth column are unique. (Property 2)

By inspection of (1), the value of the (m,q)th term of $\mathcal{H}$ is the sum of the number of paths that traverse from state j_0 to state j_{N-1} that traverse the link $j_0 \to j_1$ at at time u_m and that can traverse the link $j_{N-1} \to j_N$ at time w_q because the arrival time v_n at state j_{N-1} is less than w_q. Since these account for all the temporal paths from state j_{N-1} to j_N, (1) is the total number of paths from j_0 to j_N that traverse the link $j_0 \to j_1$ at at time u_m and that traverse the link $j_{N-1} \to j_N$ at time w_q. (Property 3) ∎

Now consider the use of this technique to calculate all the end-to-end-differential delays for the network in Fig. 2:

$$\mathcal{A}_{0,1} \odot \mathcal{A}_{1,2} \odot \mathcal{A}_{2,3} = \left[\begin{array}{cc} 2s^5 & 3s^9 \\ s^4 & 2 * s^8 \\ s^3 & 2 * s^7 \\ \hline s^6 & s^{10} \\ s^{-6} & s^{-10} \end{array} \right]$$

By inspecting this result, we see that for the output link at time 6, there are two paths of delay 5, corresponding to 1 →2→ 6 and 1 →5→ 6. There are two paths with delay 7: 3 →5→ 10 and 3 →8→ 10.

Because the entries of a temporal adjacency matrix that correspond to a single link require a lot of space and are more difficult to parse than the underlying information, we introduce the following convention. For a link that is up at times ℓ, m, and n, we write $\tilde{\mathcal{A}}_{i,j} = \tilde{\mathcal{A}}(\ell, m, n)$ to denote that

$$\mathcal{A}_{i,j} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline s^{\ell} & s^{m} & s^{n} \\ s^{-\ell} & s^{-m} & s^{-n} \end{array} \right].$$

When the times that is link is available are increased, the size of the matrix $\mathcal{A}_{i,j}$ increases correspondingly.

D. Extension to General (Non-linear) Temporal Networks

The results above explain how to find the elements of the temporal adjacency matrix and show how to perform computations using these entries. We are now ready to further explore the temporal adjacency matrix and define matrix multiplication for such matrices.

We define an operator ⊡ that generalizes matrix multiplication for regular matrices to temporal adjacency matrices. As in normal matrix multiplication, the (i,j)th entry depends on the entries in the ith row and the jth column. In normal matrix multiplication, the (i,k)th entry is multiplied by the (k,j)th entry and these are summed for all k. For the ⊡ operator, we take $\mathcal{A}_{i,k} \odot \mathcal{A}_{k,j}$ and we **concatenate** the columns of these outputs to preserve the information from each ⊙ operation. Because the outputs may vary in the number of rows, additional rows of zeros are added where needed to make the columns all have an equal number of rows.

We explain the procedure for calculating ⊡ through an example.

Consider the network with the following temporal availability:

$$\mathcal{A} = \left[\begin{array}{cccc} 0 & \tilde{\mathcal{A}}(1,5) & \tilde{\mathcal{A}}(2,3) & 0 \\ \tilde{\mathcal{A}}(1,5) & 0 & 0 & \tilde{\mathcal{A}}(4) \\ \tilde{\mathcal{A}}(2,3) & 0 & 0 & \tilde{\mathcal{A}}(1,6,8) \\ 0 & \tilde{\mathcal{A}}(4) & \tilde{\mathcal{A}}(1,6,8) & 0 \end{array} \right]$$

Let $\mathcal{A}^2 = \mathcal{A} \boxdot \mathcal{A}$. The overall matrix $\mathcal{A}^2$ is large, so we consider a few of the components. The value of $\mathcal{A}^2_{2,2}$ is the concatenation of the results of $\mathcal{A}_{2,0} \odot \mathcal{A}_{0,2}$ and $\mathcal{A}_{2,3} \odot \mathcal{A}_{3,2}$ because for the other $\mathcal{A}_{2,k} \odot \mathcal{A}_{k,2}$, at least one of the values is zero. Thus, the two components of $\mathcal{A}^2_{2,2}$ are

$$\mathcal{A}_{2,0} \odot \mathcal{A}_{0,2} = \begin{array}{cc} 0 & s \\ 0 & 0 \\ \hline s^2 & s^3 \\ s^{-2} & s^{-3} \end{array}$$

and

$$\mathcal{A}_{2,3} \odot \mathcal{A}_{3,2} = \begin{array}{ccc} 0 & s^5 & s^7 \\ 0 & 0 & s^2 \\ 0 & 0 & 0 \\ \hline s & s^6 & s^8 \\ s^{-1} & s^{-6} & s^{-8} \end{array}$$

Since these have different numbers of rows, the two results cannot be concatenated until additional zeros are added to the result of $\mathcal{A}_{2,0} \odot \mathcal{A}_{0,2}$, above the bottom two rows. The overall result is

$$\mathcal{A}^2_{2,2} = \begin{array}{ccccc} 0 & s & 0 & s^5 & s^7 \\ 0 & 0 & 0 & 0 & s^2 \\ 0 & 0 & 0 & 0 & 0 \\ \hline s^2 & s^3 & s & s^6 & s^8 \\ s^{-2} & s^{-3} & s^{-1} & s^{-6} & s^{-8} \end{array}$$

We can create a *differential delay polynomial* (DDP) that captures all of the differential delays between two nodes over a specified number of steps as

$$\mathrm{DDP}(\mathcal{A}_{j,k}) = \sum_{\ell} a_\ell s^\ell$$

where a_ℓ is the number of paths with differential delay equal to ℓ. The DDP can be created by summing all of the elements

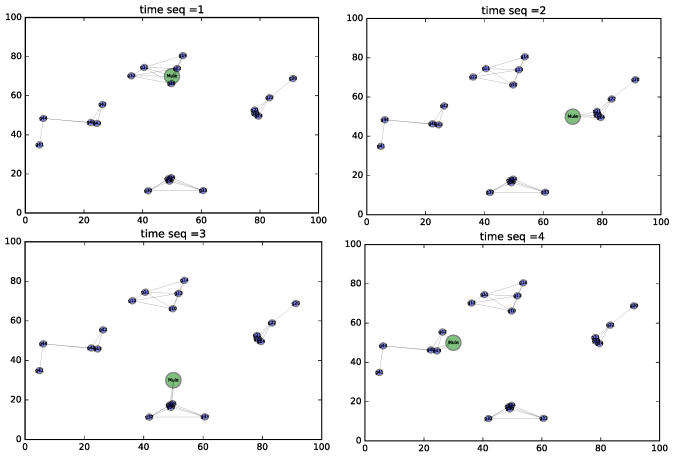

Fig. 3. Disconnected network clusters (blue nodes) served by a data mule (green node).

of the matrix entry $\mathcal{A}_{j,k}$ except for the last two rows. For instance,

$$\mathrm{DDP}(\mathcal{A}^2_{2,2}) = s + s^2 + s^5 + s^7,$$

indicating that there is exactly one path from node 2 back to node 2 for each of the differential delays, 1, 2, 5, and 7. We define the *differential delay matrix* (DDM) of a temporal adjacency matrix as the matrix $\mathcal{M} = \mathrm{DDM}(\mathcal{A})$ such that $\mathcal{M}_{j,k} = \mathrm{DDP}(\mathcal{A}_{j,k})$. Applying these rules to $\mathcal{A}^2$, we have $\mathrm{DDM}(\mathcal{A}^2) =$

$$\begin{bmatrix} s^4+s & 0 & 0 & s^6+s^5+s^4+2s^3 \\ 0 & s^4 & s^4+2s^2+s & 0 \\ 0 & 2s^3+s^2 & s^7+s^5+s^2+s & 0 \\ s^2+2s & 0 & 0 & s^7+s^5+s^2 \end{bmatrix}.$$

Thus, for instance, considering only two-hop paths between nodes 0 and 3, there are two temporal paths with differential delay 3 and one temporal path for each of the differential delays 4, 5, and 6. The differential delays for temporal paths of length 3 are completely characterized by $\mathrm{DDM}(\mathcal{A}^3) =$

$$\begin{bmatrix} 0 & s^3 & s^7+s^6+2s^5 & 0 \\ s^2 & 0 & 0 & 2s^7+2s^5+s^4 \\ s^4 & 0 & 0 & s^7+s^6+s^4 \\ 0 & 2s^4 & s^7+s^2 & 0 \end{bmatrix}.$$

IV. Example Application to Delay Tolerant Network

We consider a network that consists of four clusters of randomly scattered nodes. The clusters cannot communicate but are served by a data mule that travels between the clusters to carry data between them, as shown in Fig. 3. Because of space limitations, we only consider the first 4 steps of the data mule's travel among the nodes. For each pair of nodes, we use the methods of Section III to determine the number of walks of length 1, 2, and 3 that exist between those nodes. In the heat maps in Fig. 4, the (i, j)th pixel color indicates indicate the log of the number of walks (red=most, dark blue=0) connecting node i to node j at the specified delay. The results show that as the data mule progresses around the network, it increases the connectivity significantly, but additional travel around the network is needed to reach full connectivity. Moreover, different pairs of nodes have very different levels of connectivity.

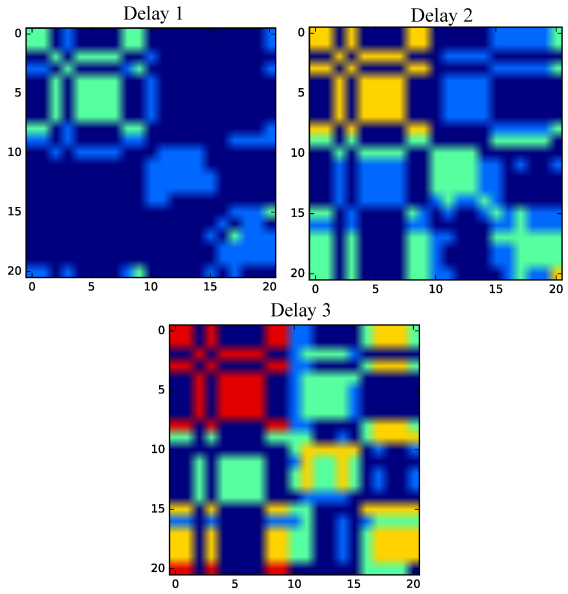

Fig. 4. Visualization of number of paths connecting nodes.

V. Conclusion

In this paper, we presented a new method for computing the possible delays for communication among nodes in a temporal network. We extended the idea of adjacency matrices for graphs to handle the time-varying nature of temporal networks. The techniques are appropriate for implementation on computers and can be used to produce visualizations that give insight into network dynamics.

References

[1] P. Holme and J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[2] P. Holme, *Temporal networks*. Springer, 2014.

[3] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, "Graph metrics for temporal networks," in *Temporal networks*. Springer, 2013, ch. 2, pp. 15–40.

[4] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia, "Analysing information flows and key mediators through temporal centrality metrics," in *Proceedings of the 3rd Workshop on Social Network Systems*, 2010, p. 3.

[5] R. K. Pan and J. Saramäki, "Path lengths, correlations, and centrality in temporal networks," *Physical Review E*, vol. 84, no. 1, p. 016105, 2011.