

Structural controllability of temporal networks

Márton Pósfai^{1,2} and Philipp Hövel^{2,3}

¹ Department of Physics of Complex Systems, Eötvös University, Pázmány Péter sétány 1/A, H-1117 Budapest, Hungary

² Institut für Theoretische Physik, Technische Universität Berlin, Hardenbergstraße 36, 10623 Berlin, Germany

³ Bernstein Center for Computational Neuroscience, Humboldt-Universität zu Berlin, Philippstraße 13, 10115 Berlin, Germany

E-mail: posfaim@elte.hu

Received 13 October 2014

Accepted for publication 13 November 2014

Published 22 December 2014

New Journal of Physics **16** (2014) 123055

doi:[10.1088/1367-2630/16/12/123055](https://doi.org/10.1088/1367-2630/16/12/123055)

Abstract

The control of complex systems is an ongoing challenge of complexity research. Recent advances using concepts of structural control deduce a wide range of control related properties from the network representation of complex systems. Here, we examine the controllability of systems for which the timescale of the dynamics we control and the timescale of changes in the network are comparable. We provide analytical and computational tools to study controllability based on temporal network characteristics. We apply these results to investigate the controllable subnetwork using a single input. For a generic class of model networks, we witness a phase transition depending upon the density of the interactions, describing the emergence of a giant controllable subspace. We show the existence of the two phases in real-world networks. Using randomization procedures, we find that the overall activity and the degree distribution of the underlying network are the main features influencing controllability.

Keywords: complex systems, temporal networks, structural controllability



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

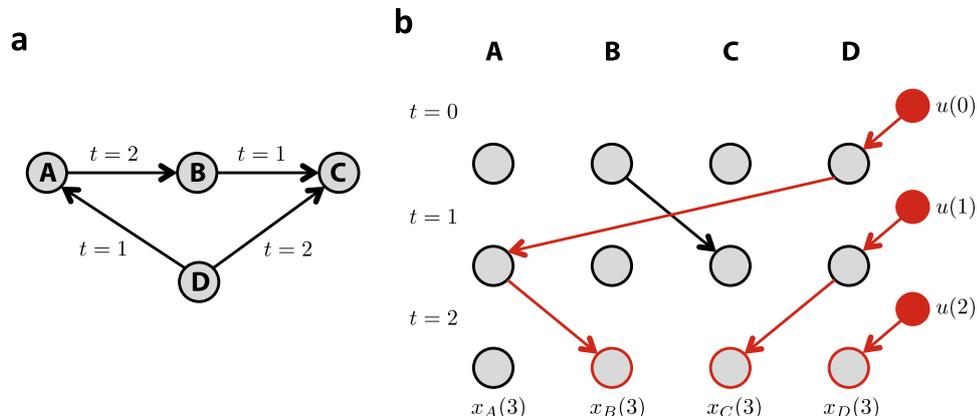


Figure 1. Controlling a simple temporal network. (a) In the aggregated network, information can spread from node A to C, however, it is forbidden if the temporal order of the interactions is taken into account. As a consequence, we cannot control C by imposing a control signal on A. (b) We visualize the dynamics represented by the small temporal network by creating a copy of each node for each time step. The state of the node in the layer $t + 1$ is determined by its neighbors in layer t . We aim to control the system at $t = 3$ in $\Delta t = 3$ time steps. We use D as an input node, meaning that we can intervene at D in layers $t = 1, 2, 3$. According to the independent path theorem, we can control the nodes in layer $t = 3$ that can be connected to intervention points via independent paths; therefore we control nodes B, C, and D.

1. Introduction

Complex systems consist of many interacting elements, and the web of these interactions is best described as a complex network. Therefore, studying the structure of such networks and exploring the consequences of their properties is essential to understand complexity. In the last two decades, a significant amount of research has been devoted to this problem [1–4], however, only limited progress has been made in describing how the network structure of a system influences our ability to control it [5–8]. Recent work by Liu *et al* spurred interest in network control [9]. They found that, if the system can be represented by a directed weighted network, assuming linear dynamics and invoking the framework of structured systems [10, 11], it is possible to study control-related questions by only using information about the underlying network. This enabled the research community to apply the full arsenal of network science to the problem, uncovering various nontrivial phenomena emerging from the complexity of the structure of the system [12–17].

In this paper, we extend structural controllability to systems for which the timescale of the dynamics and the timescale of changes in the network topology are comparable [18]. In particular, it is necessary to take into account temporal information of the connections when the interaction events are not evenly distributed over time, but have nontrivial temporal correlations [19, 20]. Such systems include communication, trade, or transportation networks [21–25]. Furthermore, the temporal sequence of interactions governs spreading processes [21, 22]. Consider as an example a small communication network of three individuals A, B, and C (figure 1(a)). Assume that B sends an email to C at time $t = 1$, and A sends an email to B at time $t = 2$. Neglecting the temporal sequence, we find that information may spread from A to C. However, taking the order of the messages into account, this is obviously not possible, which

has a clear consequence for control: we cannot influence C using A . Therefore, one must include the temporal aspect of the interactions when studying the controllability of networks with time-varying topologies.

2. Structural controllability of temporal networks

We study directed temporal networks \mathcal{T} , which are composed of a set of nodes $V = \{v_1, v_2, \dots, v_N\}$ and a set of temporal links $E = \{e_1, e_2, \dots, e_L\}$. Each temporal link $e = (v_i, v_j, w_{ij}, t) \in E$ consists of an ordered node pair and a time stamp, representing that the node v_i interacts with node v_j at time t , and w_{ij} is the link weight. In a static network the weight sometimes represents the frequency of interaction; here it describes the strength of an individual interaction. For example, in a social network w_{ij} may measure how much v_j trusts v_i . For our purposes, the detailed value of w_{ij} does not have to be known; therefore we refer to link $e = (v_i, v_j, w_{ij}, t)$ as $e = (v_i, v_j, t)$.

We consider discrete time-varying linear dynamics [26]

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (1)$$

where the vector $\mathbf{x}(t) \in \mathbb{R}^N$ represents the state variables, $x_i(t)$ corresponding to the state of node v_i at time t . The first term describes the internal dynamics of the system, the matrix $\mathbf{A}(t) \in \mathbb{R}^{N \times N}$ is the transpose of the weighted adjacency matrix at time t . The second term describes the control applied to the system: if we impose an outside signal on node v_i at time t changing the state of the node at time $t+1$, we say that we intervene at node v_i , and we call the $(v_i, t+1)$ pair an intervention point. The vector $\mathbf{u}(t) \in \mathbb{R}^{N_I(t)}$ is a list of interventions, where $N_I(t)$ is the number of interventions at time t . The nonzero elements of matrix $\mathbf{B}(t) \in \mathbb{R}^{N \times N_I(t)}$ identify the intervention points.

Extending the standard definition of structural controllability to time-varying systems [10, 27], we call a subset of nodes $C \subseteq V$ a structural controllable subspace at target time t in Δt time steps, if there exists a pair of $\mathbf{A}^*(t)$ and $\mathbf{B}^*(t)$ that has the same structure as $\mathbf{A}(t)$ and $\mathbf{B}(t)$, such that the state of all nodes $v \in C$ can be driven from any initial state to any final state at time t , in at most Δt time steps by appropriately choosing $\mathbf{u}(t)$. By the same structure we mean that the zero entries in $\mathbf{A}(t)$ and $\mathbf{A}^*(t)$ are in the same places and only the value of the nonzero elements can be different, i.e. the links connect the same nodes in the corresponding network, and only the weights can be different. It is worth noting that explicitly including Δt in the definition allows us to study the time necessary to achieve control, an aspect that has not been explored yet.

The power of the structural controllability approach arises from the fact that it does not require detailed information about the strength of the interactions, allowing us to characterize controllability by considering network properties only. Yet, structural controllability is a general property, in the sense that if a system is structurally controllable, it is controllable for almost all weight configurations [10, 11].

We prove the independent path theorem in appendix A. The theorem states that C is a structurally controllable subspace, if all nodes $v \in C$ at time t are connected to intervention points through independent time-respecting paths of length of at most $\Delta t - 1$. A time-respecting path is a sequence of adjacent temporal links \mathcal{T} such that subsequent links in the path are active in subsequent time steps, e.g. the link (v_i, v_j, t) may be followed by $(v_j, v_k, t+1)$. Two paths are

independent if they do not pass the same node at the same time. For a small example see figure 1.

The independent path theorem allows us to formulate control related questions. Here we focus on the problem of identifying the maximum controllable subspace $N_C(v, t, \Delta t)$ using a single input node v , i.e. we allow interventions at points (v, s) for any $t - \Delta t < s \leq t$. To determine $N_C(v, t, \Delta t)$, we have to find the maximum number of independent paths starting from possible intervention points and ending at time t , i.e. ending at points (w, t) for any $w \in V$. Identifying the independent paths can be done efficiently using the Ford–Fulkerson algorithm as explained in appendix A.6. We characterize the overall controllability of a temporal network by the average maximum controllable subspace

$$N_C(t, \Delta t) = \frac{1}{N} \sum_{v \in V} N_C(v, t, \Delta t). \quad (2)$$

3. Analytical results for model networks

We provide an analytical solution for a simple class of model networks to gain insight on the effect of the degree distribution and the choice of Δt . To create the network, we generate an uncorrelated static network for each time step with prescribed in- and out-degree distributions $p_{\text{in}}(k)$ and $p_{\text{out}}(k)$, respectively. Each time step is generated independently; only the degree distributions are kept the same. Therefore, consecutive time steps are completely uncorrelated. Since all time steps are statistically equivalent, the maximum controllable subspace does not depend on t , i.e. $N_C(t, \Delta t) \equiv N_C(\Delta t)$. We study networks with Poisson (Erdős–Rényi networks) and scale-free degree distributions, the latter meaning that $p_{\text{in/out}}(k) \sim k^{-\gamma_{\text{in/out}}}$.

Consider first the case of only one intervention point (v, s) . We can use this intervention to control one of the accessible nodes at later time steps, i.e. any node that can be reached via a path originating from (v, s) . The cluster of accessible nodes can be described as generated by the Galton–Watson branching process [45]: node v at time s has k_{out} offspring and k_{out} is drawn from the distribution $p_{\text{out}}(k)$; each of these offspring have k_{out} out-neighbors also drawn independently from $p_{\text{out}}(k)$, and so forth. The Galton–Watson process undergoes a phase transition depending on the average degree: in the subcritical phase $\langle k \rangle < 1$, it will terminate in finite steps, reaching only a finite number of nodes. In the supercritical phase $\langle k \rangle > 1$, the branching process may continue forever, spanning a finite fraction of the network. We will show in the following that the existence of infinite long paths fundamentally changes the controllability of the system.

In the subcritical regime, we find that

$$N_C(\Delta t) = 1 + \sum_{t=0}^{\Delta t-2} 1 - P(t), \quad (3)$$

where $P(d)$ is the cumulative distribution function of the maximum path length originating from an intervention point. $P(d)$ is determined using a self-consistent recursive formula, and only depends on the out-degree distribution $p_{\text{out}}(k)$. For long control times $\Delta t \rightarrow \infty$, (3) predicts $N_C(\infty) = 1 + \langle d \rangle$, which is simply one larger than the average maximum path length. For the same average degree, $\langle d \rangle_{\text{ER}}$ of Poisson distributed networks is always larger than $\langle d \rangle_{\text{SF}}$ in scale-free networks. Furthermore, in the scale-free case $\langle d \rangle_{\text{SF}}$ decreases as the degree exponent γ_{out}

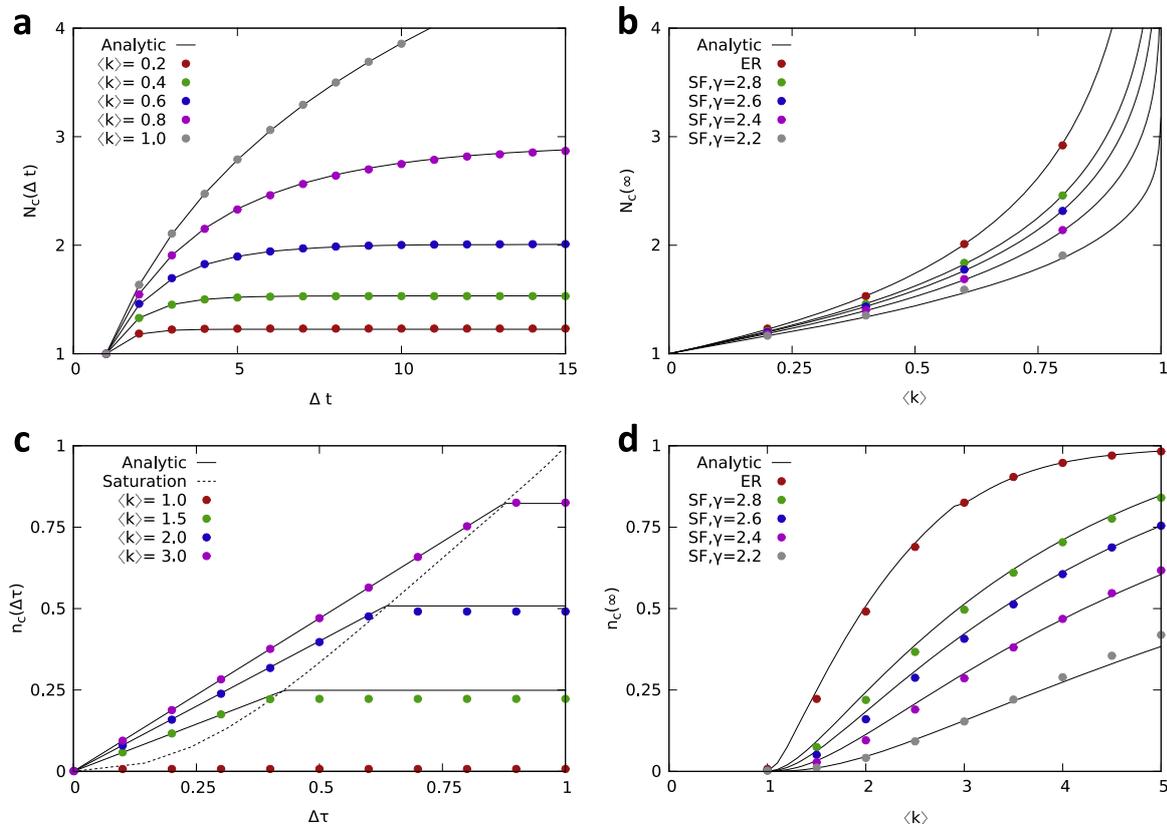


Figure 2. Temporal controllability of model networks. (a) The average maximum controllable subspace $N_C(\Delta t)$ for Poisson distributed networks in the subcritical phase. $N_C(\Delta t)$ reaches its stationary value exponentially fast, and increasing the average degree increases both $N_C(\infty)$ and the time necessary for saturation. In the critical point $\langle k \rangle = 1$, $N_C(\Delta t)$ does not saturate for finite Δt , but increases logarithmically. (b) We show the stationary value of the average maximum controllable subspace $N_C(\infty)$ in function of the average degree for Poisson and scale-free networks in the subcritical phase. The existence of hubs makes control increasingly difficult, the difference between Poisson and scale-free networks is most prominent in the critical point, where $N_C(\infty)$ diverges for Poisson distributed networks, but remains finite for scale-free networks. (c) In the supercritical phase, we can control a finite fraction of the network. We show $n_C(\Delta\tau) = N_C(\Delta\tau N)/N$ for Poisson networks. For small $\Delta\tau$, $n_C(\Delta\tau)$ increases linearly, saturates at a characteristic $\Delta\tau^*$ (dashed line), and remains constant for larger $\Delta\tau$. (d) We plot the $n_C(\infty)$ for Poisson and scale-free networks, and again find that heterogeneity of the degree distribution lowers $n_C(\infty)$. The dots are results of simulations for networks of size $N = 1\,000$; each data point is an average of 10 000 input node measurements in the subcritical phase, and 1 000 in the supercritical phase. The continuous line shows the analytic solution, for the scale-free networks we have taken the finite size effect into account (appendix B).

decreases, evincing that the presence of hubs makes control increasingly difficult. The infinite solution is approached exponentially fast for both Poisson and scale-free networks with characteristic time $\Delta t^* = 1/|\log \langle k \rangle|$. This means that only a few time steps are needed for maximum controllability (figures 2(a) and (b)).

In the transition point $\langle k \rangle = 1$, $\langle d \rangle_{\text{ER}}$ diverges, and $N_{\text{C}}^{\text{ER}}(\Delta t) \sim \log \Delta t$ for large Δt . For scale-free networks $\langle d \rangle_{\text{SF}}$ remains finite, but the asymptotic solution is reached slower, i.e. $N_{\text{C}}^{\text{SF}}(\infty) - N_{\text{C}}^{\text{SF}}(\Delta t) \sim \Delta t^{-(3-\gamma_{\text{out}})/(\gamma_{\text{out}}-2)}$. Finite size, however, can obscure the difference between the two network classes by introducing a cutoff in the degree distribution.

Above the critical point $\langle k \rangle > 1$, the maximum path length is no longer a limitation due to the formation of a giant component. Consequently, choosing large Δt , i.e. $\Delta t = \Delta\tau N$ ($\Delta\tau > 0$), we can control a finite fraction of the network $n_{\text{C}}(\Delta\tau) = N_{\text{C}}(\Delta\tau N)/N$.

For small $\Delta\tau$, any infinite path starting from an intervention point can be used for control. Therefore,

$$n_{\text{C}}(\Delta\tau) = \Delta\tau S_{\text{out}}, \quad (4)$$

where S_{out} is the probability that an intervention point is a root of an infinite path, which is determined by a self-consistent equation. Similarly to the subcritical regime, the solution only depends upon $p_{\text{out}}(k)$.

Examining (4), one might think that by allowing sufficiently large $\Delta\tau$, we can control the entire network. However, above a characteristic $\Delta\tau^*$, a new limitation arises, and $n_{\text{C}}(\Delta\tau)$ saturates (figure 2(c)). The number of controlled nodes will be equal to the maximum number of independent infinite paths, i.e. infinite paths that do not pass the same node at the same time step. We analytically approximate $n_{\text{C}}(\infty)$ using the framework developed to study core percolation and maximum matching in appendix B [29, 30]. We find that $n_{\text{C}}(\infty)$ depends on both $p_{\text{in}}(k)$ and $p_{\text{out}}(k)$, and it is symmetric to swap the two distributions: it does not matter which direction we follow the paths, the number of independent paths remains the same.

Comparing the Poisson and scale-free distributions, we find that the Poisson distributed networks are easier to control both below and above the saturation point $\Delta\tau^*$, in line with our observation in the subcritical regime (figures 2(b) and (d)).

4. Temporal controllability of a real system

Digital traces of communication make it possible to apply the developed tools to explore the controllability of real systems. Linear dynamics may provide a simple model for opinion formation: the opinion of each individual is represented by a continuous variable, and it is re-evaluated in each time step depending on the information received. The information is taken into account with different weights depending on the source [31, 32].

Here, we study a temporal network representing the email communication of a mid-size company [48]. The dataset contains the sender, the recipient and the time each email has been sent. All together there are 82 927 emails between 167 employees covering a 9 month period. The necessary temporal resolution of the network depends on the timescale of the dynamical process we aim to control. To highlight different features of the dataset we use two different temporal resolutions with one hour and one day time steps. The first corresponds to a short-term control scenario, influencing the dynamics within a workday, while the second case assumes a slower change, spanning the whole available period. The dataset features strong daily and weekly patterns: the bulk of the email traffic happens during a 9 hour period of each workday. Therefore, for the short-term control case we average the results for workdays only, and for the long-term case we remove the weekends and holidays.

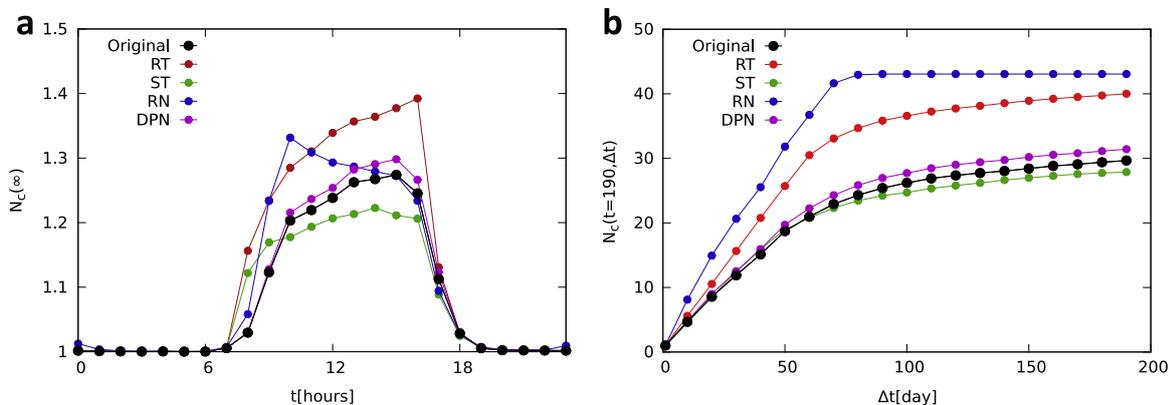


Figure 3. Controllability of a real system. (a) In the first scenario, we aim to control the company within a workday, thus we set the temporal resolution to 1 hour. The average degree of the network within a time step is $\langle k \rangle_h \approx 0.23 < 1$, that is, the system is in the subcritical phase. We show $N_C(t, \Delta t = \infty)$, i.e. the average maximum controllable subspace as a function of the target time t in the $\Delta t \rightarrow \infty$ limit. The data points are the average of 189 workdays covered by the dataset. The bulk of the email traffic happens during the working hours. Therefore, we restrict ourselves to the regular office hours spanning the period $9 \leq t \leq 17$ when randomizing the time of the links. Completely removing temporal patterns by assigning random times to the links (RT), and completely removing the network structure by randomly placing the links within a time step (RN) both increase $N_C(t, \infty)$. Shuffling the time steps (ST) only slightly increases, and randomizing the links while keeping the degree sequence (DPN) slightly decreases $N_C(t, \infty)$. This shows that the controllability is mainly determined by the degree distribution and the overall activity pattern, and that the correlations have a smaller impact. (b) In the second scenario, we aim to control the system on a longer timescale; we chose 1 workday as a unit of time. In this case the average degree within a time step is $\langle k \rangle_d \approx 1.76 > 1$, predicting that the system is above the critical point, and that the characteristic time to control the system will be in the order of the system size. Therefore, we cannot make multiple independent measurements, and we will focus on control at the end of the last workday at $t = 190$. We show the $N_C(t = 190, \Delta t)$ in function of Δt . We observe linear growth for small Δt and saturation for large Δt , although the saturation does not happen completely in the available time period. Randomizations yield similar conclusions as in the short-term control scenario.

The average degree of the network in one time step depends on the time resolution. For the one hour time steps we find $\langle k \rangle_h \approx 0.23$, predicting that the system is in the subcritical phase. Indeed, we find that $N_C(t, \Delta t)$ remains of the order of few nodes, and it saturates in just a few steps in accordance with our findings for model networks (figure 3(a)). For the one day time step, we obtain $\langle k \rangle_d \approx 1.76$, putting the system in the supercritical regime. We expect the saturation time to be in the order of the system size; therefore to minimize the effect of boundary conditions, we only consider control at the end of the last available workday, at target time $t = 190$. We find that in the beginning $N_C(t, \Delta t)$ increases approximately linearly with Δt , and for larger Δt it seems to saturate, although slower than in the case of the model networks (figure 3(b)).

Next, we use various randomization processes to separate the effects of temporal patterns and the underlying network. We find that fluctuations in the average degree of the time steps decreases $N_C(t, \Delta t)$: a drop in the average degree acts as a bottleneck, letting through fewer

independent paths. Indeed, removing the fluctuations by assigning random times to the links significantly increases $N_C(t, \Delta t)$ (RT curve in figure 3). Next, we shuffle the order of all the time steps and leave the network structure within a time step unchanged. This way we keep the overall fluctuations in the average degree, but we eliminate the correlations between subsequent time steps. We find that $N_C(t, \Delta t)$ slightly decreases, suggesting that temporal correlations enhance the number of available paths, such as casual chain of events (ST curve). To investigate the effect of the underlying network, we keep the temporal information, and we only randomize the network within a time step. First, we completely mix the connections, thereby transforming the degree distribution to a Poisson distribution with the same average degree. The controllability of the resulting network dramatically increases, showing that the existence of hubs makes control difficult (RN curve). In the next randomization, we keep the degree of each node in each time step, but we eliminate all other correlations by cutting all links and randomly rewiring them. We find that the controllability of such networks is very close to the original, meaning that the degree sequence of the nodes is the main factor in determining controllability, and that correlations are only secondary (DPN curve). For details on the dataset and the randomization procedures see appendix C.

5. Discussion

Both structural controllability and temporal networks proved to be a useful tool in understanding complex systems, generating a high amount of research in their respective fields. We have established the connection between the two, opening an array of new questions.

For instance, the effect of self-interactions [34, 35]. Self-loops can be used to model memory effects, meaning that a node remembers incoming signals for a given time. The tools provided in this paper are applicable regardless of the presence of self-loops. However, when analyzing model networks and empirical data, we made the choice not to add self-interactions explicitly.

Another important question is the problem of control energy: it has been found for static networks that even if the system is controllable, in some cases the amount of energy necessary to control the system can be very large [13]. It would be interesting to know how control energy depends on the temporal characteristics of the system, and how the intervention points should be chosen to minimize this energy.

Further possible research includes identification of the minimum set of input nodes necessary for complete control, or characterizing the role of individual nodes.

Our work focused on network effects by assuming linear dynamics; however, nonlinear dynamics are an important part of complexity. Therefore developing nonlinear control methods for complex networks is of utmost interest [36].

Temporal networks are an important ingredient in developing control methods for complex systems. Our work contributes to uncovering this rich and unexplored field and we hope that it will serve as a starting point for future investigations in this emerging area of research.

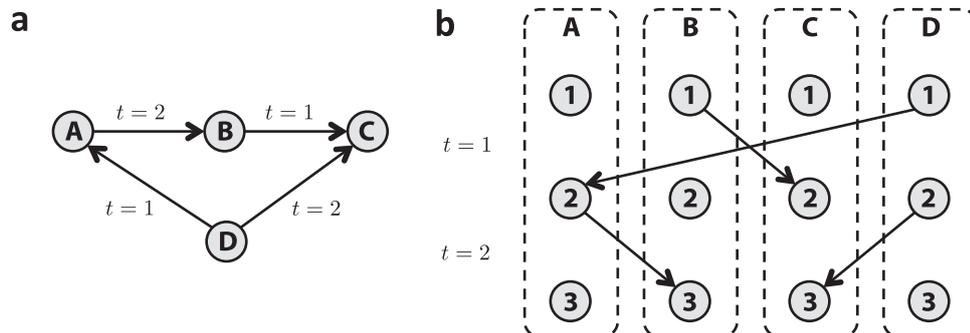


Figure A1. Layered network example. (a) A small temporal network of 4 nodes. In the static representation of the network, information can spread from node A to node C, however, due to the temporal sequence of the interactions, this is not possible. There is a time-respecting path from D to B (consisting of links $(D, A, 1)$ and $(A, B, 2)$), and from D to C (consisting of link $(D, C, 2)$). The two paths do not pass the same node at the same time; therefore they are independent. (b) The layered network representation $\mathcal{L}(t_0 = 0, t_1 = 3)$. We make a copy $\hat{v}_{(i,t)}$ of each node v_i for each time step $t \in (t_0, t_1]$. We connect the two nodes $\hat{v}_{(i,t)}$ and $\hat{v}_{(j,t+1)}$ if there exists a temporal link connecting nodes v_i and v_j at time t . There are altogether $|\hat{V}| = |V|(t_1 - t_0) = 12$ nodes in $\mathcal{L}(0, 3)$.

Acknowledgments

This project was supported by German Academic Exchange Service (DAAD) via a scholarship granted to MP, and PH acknowledges support by BMBF (grant no. 01Q1001B) in the framework of BCCN Berlin.

Appendix A. Structural controllability of temporal networks

A.1. Temporal networks

A directed temporal network \mathcal{T} is defined as a set of nodes $V = \{v_1, v_2, \dots, v_N\}$ and a set of temporal links $E = \{e_1, e_2, \dots, e_L\}$ [18]. Each temporal link $(v_i, v_j, w_{ij}, t) \in E$ consists of an ordered node pair and a time stamp, representing that there is a link pointing from node v_i to node v_j at time t with weight w_{ij} . We measure the time in discrete steps $t = 0, 1, 2, \dots$, and the choice of the unit may depend on the resolution of the available dataset or modelling purposes. For our purposes, the detailed value of w_{ij} does not have to be known, therefore we refer to link $e = (v_i, v_j, w_{ij}, t)$ as $e = (v_i, v_j, t)$.

If there exists a link $(v_i, v_j, t) \in E$, then v_i is the in-neighbor of v_j , and v_j is the out-neighbor of v_i . The temporal links (v_i, v_j, t) and (v_k, v_l, s) are consecutive, if $j = k$ and $t + 1 = s$. A temporal path P connecting node v_i and v_j from t_0 to t_1 is a sequence of consecutive temporal links such that the first link originates from node v_i at time $t_0 + 1$, and the last link in the sequence points at node v_j at time t_1 . The path consists of $\Delta t = t_1 - t_0$ consecutive links and $\Delta t + 1$ nodes. A node by itself is a path of length 0. Two paths are independent if they do not pass through the same node at the same time. For a small example see figure A1 (a).

Note: this definition of time-respecting path does not exclude waiting at nodes; waiting can be realized by adding self-loops.

A.2. Layered network representation

It will be useful to represent the temporal network defined above as a layered network $\mathcal{L}(t_0, t_1)$ consisting of a set of nodes \hat{V} and a set of static links \hat{E} . We make a copy $\hat{v}_{(i,t)} \in \hat{V}$ of each node $v_i \in V$ for each time step $t \in (t_0, t_1]$. We connect the two nodes $\hat{v}_{(i,t)}$ and $\hat{v}_{(j,t+1)}$ if there exist a temporal link (v_i, v_j, t) (see figure A1(b)). Therefore, the layered representation is a static directed acyclic network with $|\hat{V}| = |V|(t_1 - t_0)$ nodes.

As a consequence, temporal paths appear as static paths in the layered representation, and independent temporal paths are simply node-disjoint paths.

A.3. Dynamics

We study discrete time linear dynamics [26]

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (\text{A.1})$$

where the vector $\mathbf{x}(t) \in \mathbb{R}^N$ represents the state with variables, with $x_i(t)$ corresponding to the state of node v_i at time t . The matrix $\mathbf{A}(t) \in \mathbb{R}^{N \times N}$ provides information about the interactions at time t : if there exists a link (v_i, v_j, w_{ij}, t) , the matrix element $a_{ji}(t) = w_{ij}$, otherwise $a_{ji}(t) = 0$. The vector $\mathbf{u}(t) \in \mathbb{R}^{N_I(t)}$ is the control signal, we call each element of $\mathbf{u}(t)$ an intervention, and $N_I(t)$ is the number of interventions at time t . The matrix $\mathbf{B}(t) \in \mathbb{R}^{N \times N_I(t)}$ tells us at which node we intervene: $b_{ij}(t) \neq 0$ means that we shift $x_i(t+1)$ by $b_{ij}(t)u_j$. We define the total number of interventions $N_I = \sum_t N_I(t)$. If we intervene at a node at any time, the node is referred to as an input.

Note 1: the state $x_i(t+1)$ of node v_i is completely determined by the state of its in-neighbors at time t . If we assume that $x_i(t+1)$ is not independent from $x_i(t)$, we have to add self-interactions (e.g. diagonal entries in $\mathbf{A}(t)$). Information about self-interactions is not always explicitly provided in network datasets.

Note 2: consider the case when there are no links pointing at v_i at t , that is $a_{ij}(t) = 0$ for all j . If we did not add self-interactions, then $x_i(t+1) = 0$. In some cases, we might assume that if a node does not have incoming links, it retains its state. This can be taken into account by adding a self-interaction at time t only if a node has no incoming links.

A.4. Controllability

Controllable: we call the system $(\mathbf{A}(t), \mathbf{B}(t))$ controllable at target time t_1 in Δt time steps, if the system can be driven to any final state $\mathbf{x}(t_1)$ at time t_1 from any initial state $\mathbf{x}(t_0)$ in at most $\Delta t = t_1 - t_0$ time steps. Since the system is linear, we can transform $\mathbf{x}(t)$ to set $\mathbf{x}(t_0) = 0$ without loss of generality.

By successively applying (A.1), the state of the system at time t_1 is

$$\begin{aligned} \mathbf{x}(t_1) = & \mathbf{A}(t_1-1)\mathbf{A}(t_1-2)\cdots\mathbf{A}(t_0+1)\mathbf{B}(t_0)\mathbf{u}(t_0) + \dots \\ & + \mathbf{A}(t_1-1)\mathbf{B}(t_1-2)\mathbf{u}(t_1-2) + \mathbf{B}(t_1-1)\mathbf{u}(t_1-1). \end{aligned} \quad (\text{A.2})$$

We define the temporal controllability matrix [26, 27]

$$\mathbf{C}(t_0, t_1) = \left[\mathbf{A}(t_1-1)\mathbf{A}(t_1-2)\cdots\mathbf{A}(t_0+1)\mathbf{B}(t_0); \dots; \mathbf{A}(t_1-1)\mathbf{B}(t_1-2); \mathbf{B}(t_1-1) \right], \quad (\text{A.3})$$

where $[\mathbf{X}; \mathbf{Y}]$ is the concatenation of matrices \mathbf{X} and \mathbf{Y} ; therefore $\mathbf{C}(t_0, t_1) \in \mathbb{R}^{N \times N_1}$ with the total number of interventions $N_1 = \sum_t N_1(t)$. Using this definition we simply get

$$\mathbf{x}(t_1) = \mathbf{C}(t_0, t_1)\mathbf{u}, \quad (\text{A.4})$$

where $\mathbf{u} = [\mathbf{u}(t_0)^T; \mathbf{u}(t_0 + 1)^T; \dots; \mathbf{u}(t_1 - 1)^T]^T \in \mathbb{R}^{N_1}$. It is now clear that the linear rank of $\mathbf{C}(t_0, t_1)$ is the number of variables that can be set independently by the proper choice of \mathbf{u} , that is $(\mathbf{A}(t), \mathbf{B}(t))$ is controllable if

$$\text{rank } \mathbf{C}(t - \Delta t, t) = N. \quad (\text{A.5})$$

In most cases, however, the strength of the interactions, i.e. the link weights, are not known completely. Fortunately, a lot of information about the controllability of a system can be deduced only from the zero-nonzero structure of $(\mathbf{A}(t), \mathbf{B}(t))$, i.e. the existence or absence of links, using the structural controllability framework [10]. We treat the nonzero elements of $(\mathbf{A}(t), \mathbf{B}(t))$ as free parameters, and only keep the zero elements fixed.

Structurally controllable: we call the system $(\mathbf{A}(t), \mathbf{B}(t))$ structurally controllable at target time t in Δt time steps, if we can set the free parameters of $(\mathbf{A}(t), \mathbf{B}(t))$ such that the system is controllable in the original sense.

Note: if a system is structurally controllable, it is controllable for almost all weight configurations. And if it is not, it can be made controllable with an arbitrarily small perturbation of the weights [10].

Controllable subspace: we call the subset of state variables $C \subseteq V$ a controllable subspace at target time t_1 in Δt time steps, if the state variables $x_i \in C$ can be driven to any final state at target time t_1 from any initial state in at most $\Delta t = t_1 - t_0$ time steps.

Structurally controllable subspace: we call the subset of state variables $C \subseteq V$ a structurally controllable subspace at target time t in Δt steps, if we can set the free parameters of $(\mathbf{A}(t), \mathbf{B}(t))$ such that C is a controllable subspace in the original sense.

A.5. Independent path theorem

Theorem. $C \subseteq V$ is a controllable subspace of \mathcal{T} with dynamics $(\mathbf{A}(t), \mathbf{B}(t))$ at target time t_1 in Δt time steps, iff there exists $|C|$ independent paths starting from intervention points within $(t_1 - \Delta t, t_1]$ and ending at nodes $v_i \in C$ at time t_1 .

Proof. We reduce the time-dependent controllability problem to a larger time-independent problem

$$\hat{\mathbf{x}}(\hat{t} + 1) = \hat{\mathbf{A}}\hat{\mathbf{x}}(\hat{t}) + \hat{\mathbf{B}}\hat{\mathbf{u}}. \quad (\text{A.6})$$

We construct the linear time-independent system the following way. We create $\hat{\mathbf{x}} \in \mathbb{R}^{N \cdot \Delta t}$ state vector, such that $\hat{x}_{(i,t)}(\hat{t})$ corresponds to $x_i(t)$, $t \in (t_1 - \Delta t, t_1]$. Note that we use the index pair (i, t) to identify the elements of vector $\hat{\mathbf{x}}$. We construct $\hat{\mathbf{A}}$ by setting $\hat{a}_{(i,t);(j,t+1)}$ to $a_{ij}(t)$, while all other elements of $\hat{\mathbf{A}}$ are set to 0. The input nodes in the time-independent system correspond to the intervention points of the time-dependent system, that is $\hat{\mathbf{u}} = [\mathbf{u}(t_0)^T; \mathbf{u}(t_0 + 1)^T; \dots; \mathbf{u}(t_1 - 1)^T]^T = \mathbf{u}$, and $\hat{b}_{(i,t);j}$ is $b_{ij}(t)$. The network representation of the time-independent system is equivalent to the layered graph representation $\mathcal{L}(t_1 - \Delta t, t_1)$ of the temporal network \mathcal{T} .

a

$$\mathbf{A}(1) = \begin{pmatrix} 0 & 0 & 0 & a_{AD}(1) \\ 0 & 0 & 0 & 0 \\ 0 & a_{CB}(1) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{A}(2) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a_{BA}(1) & 0 & 0 \\ 0 & 0 & 0 & a_{CD}(1) \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B}(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ b_{D1}(0) \end{pmatrix} \quad \mathbf{u}(0) = (u_1(0)) \quad \mathbf{B}(1) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ b_{D1}(1) \end{pmatrix} \quad \mathbf{u}(1) = (u_1(1)) \quad \mathbf{B}(2) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ b_{D1}(2) \end{pmatrix} \quad \mathbf{u}(2) = (u_1(2))$$

$$\mathbf{x}(3) = \mathbf{A}(2)\mathbf{A}(1)\mathbf{B}(0)\mathbf{u}(0) + \mathbf{A}(1)\mathbf{B}(1)\mathbf{u}(1) + \mathbf{B}(2)\mathbf{u}(2) = \begin{pmatrix} 0 \\ a_{BA}(2)a_{AD}(1)b_{D1}(0)u_1(0) \\ a_{CD}(2)b_{D1}(1)u_1(1) \\ b_{D1}(2)u_1(2) \end{pmatrix}$$

b

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & 0 & 0 \\ \mathbf{A}(1) & 0 & 0 \\ 0 & \mathbf{A}(2) & 0 \end{pmatrix} \quad \hat{\mathbf{B}} = \begin{pmatrix} \mathbf{B}(0) \\ \mathbf{B}(1) \\ \mathbf{B}(2) \end{pmatrix} \quad \hat{\mathbf{u}} = \begin{pmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \mathbf{u}(2) \end{pmatrix}$$

$$\hat{\mathbf{x}}(3) = \hat{\mathbf{A}}^2\hat{\mathbf{B}}\hat{\mathbf{u}} + \hat{\mathbf{A}}\hat{\mathbf{B}}\hat{\mathbf{u}} + \hat{\mathbf{B}}\hat{\mathbf{u}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{b_{D1}(0)u_1(0)}{a_{AD}(1)b_{D1}(0)u_1} \\ 0 \\ 0 \\ \frac{b_{D1}(1)u_1(1)}{0} \\ a_{BA}(2)a_{AD}(1)b_{D1}(0)u_1(0) \\ a_{CD}(2)b_{D1}(1)u_1(1) \\ b_{D1}(2)u_1(2) \end{pmatrix}$$

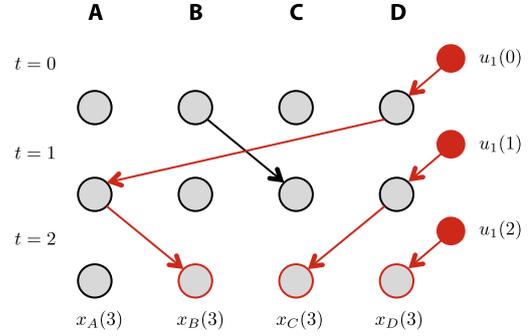
c

Figure A2. Converting time dependent dynamics to time invariant. (a) The matrices describing the linear dynamics corresponding to the example network in figure A1(a). We aim to control the system at target time $t = 3$ in $\Delta t = 3$ time steps, and we use node D as an input node. We compute $\mathbf{x}(3)$ by successfully applying (A.1). (b) We convert the linear time-varying dynamics to a larger time-independent system the following way. We create $\hat{\mathbf{x}} \in \mathbb{R}^{N \cdot \Delta t}$ state vector, such that $\hat{x}_{(i,t)}$ (\hat{t}) corresponds to $x_i(t)$, $t \in (t_1 - \Delta t, t_1]$. We construct $\hat{\mathbf{A}}$ by setting $\hat{a}_{(i,t);(j,t+1)}$ to $a_{ij}(t)$; all other elements of $\hat{\mathbf{A}}$ are set to 0. The input nodes $(D, 1)$, $(D, 2)$ and $(D, 3)$ correspond to the intervention points in the time-varying system. The state vector of the time-independent system $\hat{\mathbf{x}}(t)$ at time $t = 3$ is equal to the state vector of the time-varying system $\mathbf{x}(t)$. (c) The corresponding network of the time-independent system is equivalent to the layered graph representation $\mathcal{L}(0, 3)$ of the temporal network \mathcal{T} . According to the independent path theorem, we can control nodes B , C , and D .

We can check by simple multiplication that $\hat{x}_{(i,t_1)}(\hat{t} = \Delta t) = x_i(t_1)$ for all i . Therefore for every C controllable subspace of the system $(\mathbf{A}(t), \mathbf{B}(t))$, there exists a \hat{C} controllable subspace of $(\hat{\mathbf{A}}, \hat{\mathbf{B}})$ such that $\hat{C} \supset \{\hat{v}_{(i,t_1)} : v_i \in C\}$. It has been previously shown [11, 37] that a subspace \hat{C} of a static network is structurally controllable, if there exists a stem-cycle disjoint subgraph that contains all nodes in \hat{C} . A stem is a path starting from a node that is directly coupled to an input signal; in our case these are the intervention points. A stem-cycle disjoint subgraph is a subgraph composed of stems and cycles, such that all nodes are contained by exactly one stem or one cycle. The network representation of the time-independent system is acyclic, hence a stem-cycle disjoint subgraph in our case is simply a set of independent paths. Therefore, C is a structurally controllable subspace of $(\mathbf{A}(t), \mathbf{B}(t))$, if there exists $|C|$

independent paths starting from intervention points and leading to each node $v_i \in C$ at time t_1 . \square

For a small example see figure A2.

A.6. Maximum controllable subspace problem

Given a temporal network \mathcal{T} , we select a set of nodes $D \subseteq V$ to be inputs, meaning that we allow interventions at these nodes. We explore the problem of determining the dimension of the maximum controllable subspace $N_C(t_1, \Delta t)$, where t_1 is the target time, and Δt is the number of time steps we use to reach the desired state. We use the layered representation $\mathcal{L}(t_1 - \Delta t, t_1)$. The set of potential intervention points is $I = \{\hat{v}_{(i,t)}, \forall i, t: v_i \in D\}$ and the set of potential target nodes is $T = \{\hat{v}_{(i,t_1)}, \forall i\}$. A controllable subspace is given by a subset of T for which all nodes can be reached via independent paths from potential intervention points. Therefore, the dimension of the maximum controllable subspace $N_C(t_1, \Delta t)$ is the maximum number of independent paths originating from I and terminating in T .

The problem of finding the maximum number of independent paths in directed networks is equivalent to solving the maximum flow problem. If the nodes in set I are sources, the nodes in T are sinks, and if the capacity of each link and node is set to 1, the maximum flow is equal to the maximum number of independent paths. This problem can be solved in polynomial time, e.g. using the Ford–Fulkerson algorithm with complexity $O(|\hat{E}| \cdot N_C)$ [38, 39].

Appendix B. Analytic solution for model networks

B.1. Temporal network model definition

We study a simple uncorrelated temporal network model that can be considered as the temporal counterpart of the static hidden parameter model [40, 41]. We start with N unconnected nodes, and for each time step we generate a directed network independently. Each node v_i is assigned two hidden parameters $w_{\text{in}}(i)$ and $w_{\text{out}}(i)$. We then randomly place L directed links by choosing the start- and endpoint of the link with probability proportional to $w_{\text{in}}(i)$ and $w_{\text{out}}(i)$, respectively. By properly choosing the hidden parameters, we can tune the degree distributions $p_{\text{in}}(k)$ and $p_{\text{out}}(k)$. Throughout the paper, we investigate networks with Poisson [42] and scale-free distribution [43], the latter meaning that the distribution has a power-law tail.

The degree distribution of the model is given by

$$p(k) = \sum_{i=1}^N \exp\left[-\frac{w(i)}{\sum_j w(j)} L\right] \frac{(Lw(i)/\sum_j w(j))^k}{k!}, \quad (\text{B.1})$$

which is valid for both in- and out-degree. The corresponding generating function of this distribution is

$$G(x) = \sum_{k=0}^{\infty} p(k)x^k = \sum_{i=1}^N \exp\left[-\frac{w(i)L}{\sum_j w(j)}(1-x)\right]. \quad (\text{B.2})$$

To generate a network with Poisson distribution, we set $w_{\text{in}}(i) = 1$ and $w_{\text{out}}(i) = 1$ for all nodes. This way the probability of connecting any node pair is equal, and we recover the classic Erdős–Rényi model. The corresponding generating function is

$$G^{\text{ER}}(x) = \exp[-c(1 - x)], \quad (\text{B.3})$$

where $c = L/N = \langle k \rangle$.

To generate networks with scale-free degree distribution we use the so-called static model [44]. We set the hidden parameters of node i to $w_{\text{in/out}}(i) = i^{-\alpha_{\text{in/out}}}$, where $i = 1, 2, \dots, N$. The weights are then shuffled to eliminate any correlations. For large N , this choice yields the degree distribution

$$p^{\text{SF}}(k) = \frac{[c(1 - \alpha)^{1/\alpha}] \Gamma(k - 1/\alpha, c[1 - \alpha])}{\alpha \Gamma(k + 1)}, \quad (\text{B.4})$$

where $c = L/N = \langle k \rangle$ is equal to the average degree, and $\Gamma(n, x)$ is the upper incomplete gamma function. For large k , $p^{\text{SF}}(k) \sim k^{-(1+1/\alpha)} = k^{-\gamma}$, where $\gamma_{\text{in/out}} = 1 + 1/\alpha_{\text{in/out}}$ determines the exponent of the tail of the distribution. The corresponding generating function in the $N \rightarrow \infty$ limit is

$$G^{\text{SF}}(x) = \frac{1}{\alpha} E_{1+\frac{1}{\alpha}}[c(1 - \alpha)(1 - x)], \quad (\text{B.5})$$

where $E_n(x) = \int_1^\infty dt e^{-xt} t^{-n}$ is the exponential integral function. However, for scale-free networks we often find that finite size effects are not negligible for system sizes accessible for simulation. In these cases we have to take the finite size into account by using (B.2) explicitly.

B.2. Percolation in the temporal network model

Let us consider the case when we have only one intervention point $\hat{v}_{(i,t)}$ in the layered network. We can use this intervention to control one of the accessible nodes in a lower layer, i.e. any node that can be reached via a path originating from $\hat{v}_{(i,t)}$. The cluster of accessible nodes can be described as a cluster generated by the Galton–Watson branching process [45]: the node $\hat{v}_{(i,t)}$ has k_{out} out-neighbors, where k_{out} is drawn from the distribution $p_{\text{out}}(k)$. Each of these out-neighbors will have k_{out} out-neighbors also drawn from $p_{\text{out}}(k)$, and so forth.

We study the process in the $N \rightarrow \infty$ limit and denote the probability that the branching process continues forever by S_{out} . We can calculate S_{out} using the self-consistent equation

$$1 - S_{\text{out}} = G_{\text{out}}(1 - S_{\text{out}}). \quad (\text{B.6})$$

The equation simply means that the probability that the branching process rooted at node $\hat{v}_{(i,t)}$ stops in finite steps ($1 - S_{\text{out}}$) is equal to the probability that all branching processes rooted at each out-neighbor of node $\hat{v}_{(i,t)}$ also terminate in finite steps. The equation has a trivial solution $S_{\text{out}} = 0$. At the critical point this solution loses stability:

$$1 = G'_{\text{out}}(1) = \langle k \rangle, \quad (\text{B.7})$$

meaning that the critical point is simply determined by the average degree independent from other parameters of the degree distribution. Nodes that are roots of infinite trees form the giant out-component.

In the subcritical phase ($\langle k \rangle < 1$) the branching process will halt in finite steps, meaning that only finite number of nodes can be accessed. In the critical point ($\langle k \rangle = 1$) the size of the

largest cluster diverges; however, the relative size is still zero. In the supercritical phase ($\langle k \rangle > 1$) the branching process will continue forever with probability S_{out} .

Similarly, we calculate the probability that a randomly selected node is an offspring of an infinite cluster:

$$1 - S_{\text{in}} = G_{\text{in}}(1 - S_{\text{in}}). \quad (\text{B.8})$$

Nodes that are offspring of infinite trees form the giant in-component.

We refer to the intersection of the giant in- and out-component as the giant component.

B.3. $N_C(\Delta t)$ in the subcritical phase

The goal of this section is to determine the average $N_C(\Delta t)$ using a randomly selected node v_i as input. We start with the observation that if the network is uncorrelated, each intervention point can be treated as independently and randomly selected. In the subcritical phase the size of the accessible cluster rooted at a random node is finite. Therefore, the probability that two such clusters rooted at two randomly selected nodes overlap is 0. The probability that an intervention at $\hat{v}_{(i,t)}$ can be used to control a node at the target time is equal to the probability that a sufficiently long path is rooted at the intervention point. Hence, we first determine the cumulative distribution function of the maximum path length originating from a randomly selected point, i.e. $P(d)$ is the probability that the maximum length path originating from a node is at most d . The maximum path length from node $\hat{v}_{(i,t)}$ is one larger than the maximum path length originating from its out-neighbors. Averaging over $p_{\text{out}}(k)$ we get

$$P(d) = G_{\text{out}}(P(d-1)). \quad (\text{B.9})$$

We can solve the equation recursively starting from $P(d=0) = p_{\text{out}}(k=0)$.

Since the model is invariant to time shifts, we can set the control target time to Δt without loss of generality. The probability that an intervention at time t can be used is given by $1 - P(\Delta t - t - 1)$ if $t \neq \Delta t$, and 1 if $t = \Delta t$. Therefore we get

$$N_C(\Delta t) = 1 + \sum_{t=1}^{\Delta t-1} 1 - P(\Delta t - t - 1) = 1 + \sum_{t=0}^{\Delta t-2} 1 - P(t). \quad (\text{B.10})$$

For $\Delta t \rightarrow \infty$ we get:

$$N_C = \lim_{\Delta t \rightarrow \infty} N_C(\Delta t) = 1 + \langle d \rangle. \quad (\text{B.11})$$

We gain further insight by studying the asymptotic solution of (B.9) for models with Poisson and power-law degree distribution.

For the Poisson case $p_{\text{out}}(k)$ (or any distribution with finite variance σ_{out}^2), we can expand the generating function $G_{\text{out}}(x)$ around $x = 1$:

$$P(d) = G_{\text{out}}(1) - G'_{\text{out}}(1)(1 - P(d-1)) = 1 - \langle k \rangle(1 - P(d-1)). \quad (\text{B.12})$$

Solving the recursion we get for large d

$$1 - P(d) \sim \langle k \rangle^d. \quad (\text{B.13})$$

In other words, $P(d)$ has an exponential tail, e.g. large d values add little to $\langle d \rangle$. This means that $N_C(\Delta t)$ approximates its maximum around $\Delta t \sim \langle d \rangle$, and there is little benefit from further increasing Δt .

However, at the critical point we have $\langle k \rangle = 1$, and thus we need the second-order term in the expansion to extract the asymptotic behavior:

$$\begin{aligned} P(d) &= G_{\text{out}}(1) - G'_{\text{out}}(1)(1 - P(d-1)) + \frac{1}{2}G''_{\text{out}}(1)(1 - P(d-1))^2 \\ &= 1 - (1 - P(d-1)) + \frac{\sigma_{\text{out}}^2}{2}(1 - P(d-1))^2. \end{aligned} \quad (\text{B.14})$$

For large d , this yields

$$1 - P(d) \sim \frac{2}{\sigma_{\text{out}}^2} \frac{1}{d}. \quad (\text{B.15})$$

From this it follows that for large Δt , we get $N_c(\Delta t) \sim \log \Delta t$, meaning that increasing Δt will increase the number of nodes that we control. However, the fraction of the network that is controlled still remains 0 in the large network limit.

For scale-free networks with $\gamma < 3$, the σ_{out}^2 is infinite, and therefore the simple Taylor series expansion of the generating function in (B.14) is not sufficient. To understand the effect of a power-law distribution, we transform the generating function provided in (B.5):

$$\begin{aligned} G_{\text{out}}(x) &= \frac{1}{\alpha} [c(1-\alpha)(1-x)]^{1/\alpha} \Gamma(-1/\alpha, c[1-\alpha][1-x]) \\ &= \frac{1}{\alpha} [c(1-\alpha)(1-x)]^{1/\alpha} \left[\Gamma(-1/\alpha) - \sum_{k=0}^{\infty} \frac{(-1)^k [c(1-\alpha)(1-x)]^{-1/\alpha+k}}{k!(-1/\alpha+k)} \right]. \end{aligned} \quad (\text{B.16})$$

Using this in (B.9) and only keeping the first two terms, we get

$$\begin{aligned} P(d+1) &= \frac{1}{\alpha} [c(1-\alpha)P(d)]^{1/\alpha} \left[\Gamma(-1/\alpha) + \frac{[c(1-\alpha)P(d)]^{-1/\alpha}}{1/\alpha} \right. \\ &\quad \left. + \frac{[c(1-\alpha)P(d)]^{-1/\alpha+1}}{1/\alpha-1} \right] \\ &= 1 + cP(d) + \frac{1}{\alpha} \Gamma(-1/\alpha) [c(1-\alpha)P(d)]^{1/\alpha}. \end{aligned} \quad (\text{B.17})$$

Consider $2 < \gamma < 3$, or equivalently $1/2 < \alpha < 1$. If $c = \langle k \rangle < 1$, the asymptotic behavior of the solution is determined by the second term, and we obtain the same solution as (B.13). For the solution in the critical point $\langle k \rangle = 1$, we keep the third term, and we find

$$1 - P(d) \sim d^{-\alpha/(1-\alpha)} = d^{-1/(\gamma-2)}. \quad (\text{B.18})$$

For $2 < \gamma < 3$ this means that even in the critical point $N_c(\infty)$ will remain finite. However, $N_c(\Delta t)$ will approach its stationary value slowly, that is $N_c(\infty) - N_c(\Delta t) \sim \Delta t^{-(3-\gamma)/(\gamma-2)}$.

B.4. $n_c(\Delta \tau)$ in the supercritical phase

Above the critical point, the probability that an intervention point is a root of an infinite tree is $S_{\text{out}} > 0$, meaning that there exists infinite length paths originating from the node. As a consequence, by choosing $\Delta t = \Delta \tau N$ ($\Delta \tau > 0$) we can control a finite fraction of the network

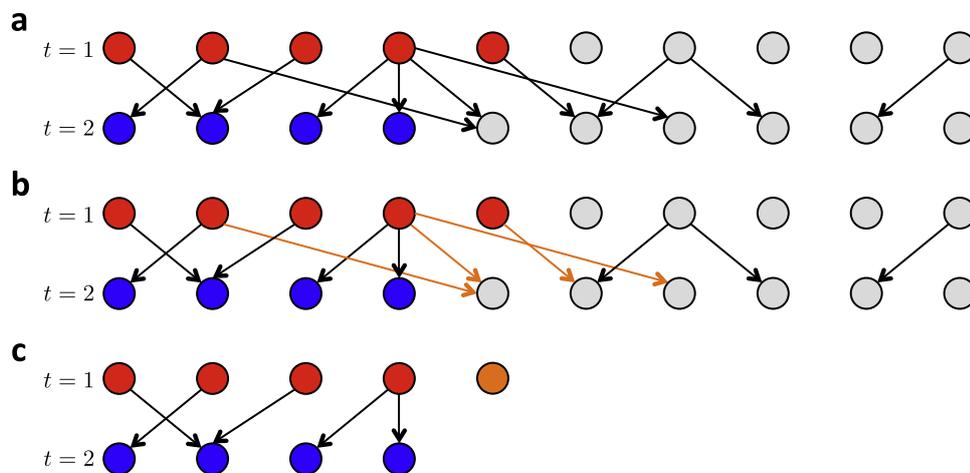


Figure B1. Determining the degree distribution within the giant component. (a) The goal is to determine the degree distribution within the giant component, i.e. nodes that belong to both the giant in-component and the giant out-component. We consider two consecutive layers $t = 1$ and $t = 2$. Assume that the red nodes are the giant in-component in layer $t = 1$, and the blue nodes are the giant out-component in layer $t = 2$. The giant component in layer $t = 1$ consists of the red nodes that are connected to at least one blue node. (b) First, we remove all links that lead to nodes outside the giant component, i.e. orange links. The network is uncorrelated, therefore the link removal can be treated as random. (c) Next, we remove the now isolated nodes (orange), i.e. nodes that are in the in-component, but not in the out-component.

$n_c(\Delta\tau) = N_c(\Delta\tau N)/N$ using infinite paths, and the contribution of finite size clusters is negligible.

Consider the case when v_i is the input node, and $v_{(i,t)}$ and $v_{(i,s)}$ are two intervention points such that both are roots of infinite trees. Since these trees cover a finite fraction of the network, we can no longer assume that the overlap of accessible clusters has zero probability. However, being the root of an infinite tree also means that we can reach a finite fraction S_{out} of the nodes in the target layer, and we can choose from many possible paths. Therefore, for small $\Delta\tau$, we assume that whenever an intervention point is a root of an infinite tree, we can use that intervention point to control one node in the target layer. This means that

$$n_c(\Delta\tau) = S_{\text{out}}\Delta\tau. \quad (\text{B.19})$$

Note that this does not depend on the in-degree distribution $p_{\text{in}}(k)$.

For large $\Delta\tau$, $n_c(\Delta\tau)$ saturates, since it is limited by the maximum throughput of the giant component, i.e. the maximum number of independent infinite paths. The giant component consists of nodes in each layer that are both in the giant in-component and the giant out-component. To calculate $n_c(\Delta\tau)$ the first step is to determine the degree distributions $\tilde{p}_{\text{in/out}}(k)$ within the giant component. Consider two adjacent layers of nodes at $t = 1$ and $t = 2$ (figure B1). The two layers are connected by links that are active at time $t = 1$. We aim to determine $\tilde{p}_{\text{out}}(k)$ for $t = 1$. The nodes that are in the giant component in layer $t = 1$ are the nodes that are in the giant in-component and have at least one connection to nodes in layer $t = 2$ that are in the giant out-component. First, we remove the links connecting the in-component with nodes not in the out-component, which is equivalent to randomly removing $1 - S_{\text{in}}$ fraction of links. Now all nodes that have at least one connection left are members of the giant component. Therefore, to

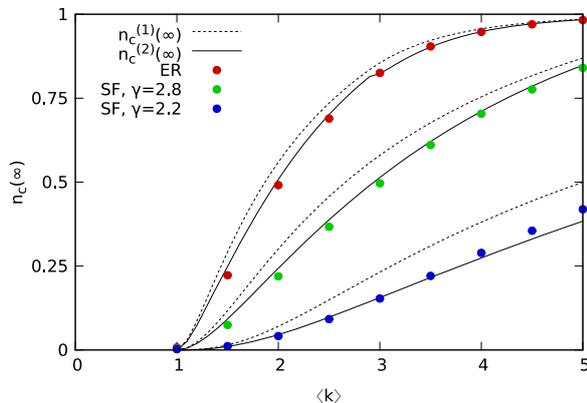


Figure B2. Comparing the approximations. We plot the average controllable fraction of the network in function of the average degree for Poisson (ER) and scale-free (SF) degree distributions. For networks of size $N = 1\,000$, each data point is an average of 1 000 input node measurements. The dashed line is the first approximation $n_C^{(1)}(\infty)$, and the solid line is the second analytical approximation $n_C^{(2)}(\infty)$. The small break in $n_C^{(2)}(\infty)$ for the Erdős–Rényi network is a consequence of the core percolation transition [29]; the transition point for scale-free networks is not shown on the plot.

obtain $\tilde{p}_{\text{out}}(k)$ we remove the nodes with 0 connections. This leads to

$$\tilde{p}_{\text{out}}(k) = \frac{1 - \delta_{k,0}}{1 - G_{\text{out}}(1 - S_{\text{in}})} \sum_{j=k}^{\infty} \binom{j}{k} S_{\text{in}}^k (1 - S_{\text{in}})^{j-k} p_{\text{out}}(j), \quad (\text{B.20})$$

and we will use the corresponding generating function

$$\tilde{G}_{\text{out}}(x) = \frac{G_{\text{out}}(1 - S_{\text{in}} + S_{\text{in}}x) - G_{\text{out}}(1 - S_{\text{in}})}{1 - G_{\text{out}}(1 - S_{\text{in}})}. \quad (\text{B.21})$$

The in-degree distribution is determined similarly.

To calculate a first approximation $n_C^{(1)}(\infty)$, we determine the maximum number of independent paths in the giant component connecting two subsequent layers $t = 1$ and $t = 2$, which is equivalent to finding the maximum matching in a bipartite network formed by the two layers. A matching in a network is defined as a set of links that do not share endpoints; therefore in the case of the network of two layers, the links in the matching are independent paths of length one. A node is called matched if it is adjacent to a link in the matching. This way $n_C^{(1)}(\infty)$ is equal to the maximum matching in a bipartite network with $S_{\text{in}}S_{\text{out}}N$ nodes in each layer, and degree distributions $\tilde{p}_{\text{out}}(k)$ and $\tilde{p}_{\text{in}}(k)$. For uncorrelated networks the size of the maximum matching can be determined analytically; we provide the detailed calculation in section B.5. This approximation yields an upper bound for $n_C(\infty)$ (figure B2), because it assumes that we can choose the endpoints of the paths in layer $t = 1$, and the starting points of the paths in layer $t = 2$ arbitrarily. However, when constructing a maximum matching we do not have such freedom: some nodes always have to be matched [15], and in other cases some nodes cannot be included at the same time. For a small example see figure B3.

For the next approximation $n_C^{(2)}(\infty)$, we consider three subsequent layers $t = 0, 1, 2$. Each layer contains $S_{\text{in}}S_{\text{out}}N$ nodes, and has degree distributions $\tilde{p}_{\text{out}}(k)$ and $\tilde{p}_{\text{in}}(k)$. First, we examine the maximum matching between layers $t = 0$ and $t = 1$, and we determine the set of nodes A in

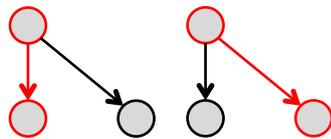


Figure B3. Role of nodes in possible maximum matchings. We show a small example network of 3 nodes. In this network the size of the maximum matching is one, and there are two possible configurations highlighted in red. The top node is always matched, and the two bottom nodes cannot be matched in the same configuration.

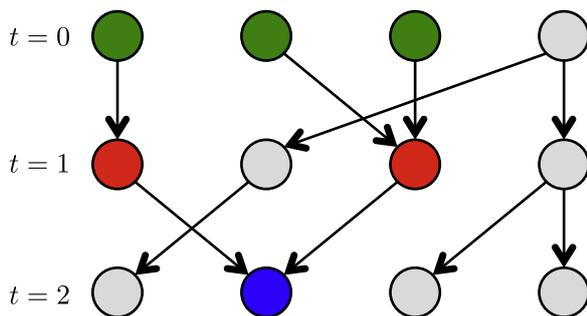


Figure B4. Second approximation of $n_C(\infty)$. Consider the giant component in three consecutive layers $t = 0, 1, 2$. Examining the maximum matching in layers $t = 0$ and $t = 1$, we find that the red nodes are matched in all possible maximum matchings. However, in the next layer the red nodes are connected to the same node, hence they cannot be matched at the same time. Therefore, the two red nodes can only be used in one independent path. Counting such configurations provides the second approximation.

layer $t = 1$ that are matched in all possible maximum matchings. In the first approximation, these nodes will always be endpoints of independent paths. However, if we cannot match them in the next layer, they will become dead ends. Therefore, the number of nodes in A that cannot be matched at the same time will be the next correction to $n_C^{(1)}(\infty)$ (figure B4). To calculate the correction, we find the maximum matching in the bipartite network formed by nodes in A in layer $t = 0$, and all nodes in layer $t = 1$. The degree distribution of nodes in A is $\tilde{p}_{\text{out}}(k)$, and the degree distribution of nodes in layer $t = 2$ can be calculated by randomly removing $1 - |A|/S_{\text{in}}S_{\text{out}}N$ fraction of links from $\tilde{p}_{\text{in}}(k)$, similarly to (B.20). The number of nodes in A is determined using the equations developed in [15].

Similar correction can be computed for the set of nodes B in layer $t = 1$ that are always matched from layer $t = 2$, but cannot be matched at the same time from layer $t = 0$. We find that $n_C^{(2)}(\infty)$ approximates the numerical simulations well (figure B2).

Note: in [15], it was shown that for dense networks above the core percolation threshold, the number of nodes that are always matched can be drastically different depending on specific realization of the network model, e.g. two Erdős–Rényi networks generated with the same parameters can be different. This is due to a special case when a finite fraction of nodes are ‘almost always’ matched, meaning that we have a set of nodes A such that in each possible matching only a finite number of nodes in A are not matched. Therefore, for our purposes these nodes can be treated as always matched.

B.5. Matching in bipartite networks

In this section we calculate the relative size of the maximum matching in uncorrelated bipartite networks with arbitrary degree distribution. Let \mathcal{B} be a bipartite network, with two sets of nodes V^- (lower) and V^+ (upper) and a set of links E , such that each link connects one upper node $v^+ \in V^+$ and one lower node $v^- \in V^-$. $p^+(k)$ and $p^-(k)$ are the degree distributions of the upper and lower sides, respectively. We use the notations $N^+ = |V^+|$, $N^- = |V^-|$ and $L = |E|$. The average degree of each layer is $c^\pm = L/N^\pm$. A matching M is the set of links that do not share endpoints; we define $m^\pm = |M|/N^\pm$. The maximum matching problem for bipartite networks has been studied for the case when $N^+ = N^-$ [9]; here we provide a solution for the $N^+ \neq N^-$ case.

We use the formalism developed for core percolation [29]. Core percolation describes the sudden emergence of the core in random networks [30, 46, 47]. To define the core, we first introduce the greedy leaf removal (GLR) process: we select a leaf randomly (a node with degree 1), and remove that node and its neighbor together with all links adjacent to that neighbor, we repeat this step until no leaves are left; we then remove all isolated nodes. The core is defined as the remainder of the network after the GLR.

Analytic description is possible by introducing the following node categories: (i) α -removable, nodes that can become isolated during the GLR; (ii) β -removable, nodes that can be removed as a neighbor of a leaf during the GLR. We define α^+ as the probability that following a random link to the upper side we find a node that is α -removable in the absence of the link. We define α^- , β^+ , and β^- similarly. These probabilities are determined by a set of self-consistent equations:

$$\alpha^\pm = H^\pm(\beta^\mp), \quad (\text{B.22})$$

$$\beta^\pm = 1 - H^\pm(1 - \alpha^\mp), \quad (\text{B.23})$$

where $H^\pm(x) = \sum_{k=1}^{\infty} k p^\pm(k) / \langle k \rangle x^{k-1}$ is the generating function of the excessive degree distribution.

The GLR process can be used to construct a maximum matching in the class of bipartite networks that we study here. We remove a leaf that consist of node v_1 with degree 1, and node v_2 with a possibly higher degree. To construct the maximum matching, we add link (v_1, v_2) to the matching. Now all links adjacent to v_2 are not allowed in the matching, and therefore we remove them, too. We can continue until we have no leaves left, i.e. we are left with the core. It was shown that in large non-bipartite random networks the core can be asymptotically matched, i.e. the probability of randomly choosing an unmatched node is zero [30]. However, in bipartite networks there is another limiting factor: if the size of the core is different on the two sides, the size of the matching in the core cannot be larger then the smaller side.

Note: this can also happen if $N^+ = N^-$, but $p^+(k) \neq p^-(k)$, and this limitation was not considered in [9] and in the subsequent [14]. Therefore, their results should be cautiously applied to networks above the core percolation with asymmetric degree distributions.

As stated above, m^\pm is the sum of the contribution of the leaf removal and the core. We first calculate the contribution of leaf removal. For each leaf removal, we add one link to the matching, increasing the number of matched nodes by 2, one on both sides. For each β -node there is one leaf removal. Therefore, to calculate the contribution of the leaf removal, we count the β -nodes on both sides:

$$N_{\beta}^{\pm} = N^{\pm} \left[1 - G^{\pm}(1 - \alpha^{\mp}) \right]. \quad (\text{B.24})$$

However, by doing this we have double counted the case when two β -nodes are removed together. This can only happen, if in the absence of the link connecting the two nodes, both nodes are α -nodes, the probability of this event is $\alpha^+\alpha^-$ for each link. Therefore, the overall contribution is

$$N^+ \left[1 - G^+(1 - \alpha^-) \right] + N^- \left[1 - G^-(1 - \alpha^+) \right] - L\alpha^+\alpha^-. \quad (\text{B.25})$$

To determine the contribution of the core, we calculate the size of the core on both sides:

$$N_{\text{core}}^{\pm} = N^{\pm} \left[G^{\pm}(1 - \alpha^{\mp}) - G^{\pm}(\beta^{\mp}) - c^{\pm}\alpha^{\pm}(1 - \beta^{\mp} - \alpha^{\mp}) \right], \quad (\text{B.26})$$

and select the smaller side. Therefore, all together we have

$$m^- = \frac{1}{N^-} \left(N^- \left[1 - G^-(1 - \alpha^+) \right] + N^+ \left[1 - G^+(1 - \alpha^-) \right] - L\alpha^+\alpha^- + \min_{\pm} N_{\text{core}}^{\pm} \right) \quad (\text{B.27a})$$

$$= \left[1 - G^-(1 - \alpha^+) \right] + \frac{N^+}{N^-} \left[1 - G^+(1 - \alpha^-) \right] - c^- \alpha^+ \alpha^- + \frac{1}{N^-} \min_{\pm} N_{\text{core}}^{\pm}. \quad (\text{B.27b})$$

Appendix C. Dataset analyzed

C.1. Description

We study a publicly available temporal network representing the email communication of a mid-sized company [48, 49]. The dataset contains the sender, the recipient, and the time each email has been sent. All together there are 82 927 emails between 167 employees covering a 9 month period.

The necessary temporal resolution of the network depends on the timescale of the dynamical process we aim to control. To highlight different features of the dataset, we use two different temporal resolutions with one hour and one day time steps. To obtain these networks, we perform a coarse graining procedure: for each time step $t_0 \leq t < t_1$ we create an aggregated network, i.e. we connect node v_i to v_j in the coarse grained network, if at least one email has been sent from v_i to v_j between t_0 and t_1 .

The one hour coarse grained network corresponds to a scenario in which we aim to influence the dynamics within a day. An important feature of the dataset is that it follows strong daily and weekly patterns. The bulk of the email traffic happens during the 9 hour period corresponding to regular office hours on workdays (figures C1(a) and (b)). The average degree of the network outside working hours is approximately 0, while during office hours $\langle k \rangle_h \approx 0.23$. This means that control on the hourly timescale is only possible within one day, that is, each day can be considered separately. We find that the average degree distribution is highly heterogeneous (figures C1(c) and (d)), and the second moment $\langle k_{\text{out}}^2 \rangle_h \approx 0.99$, $\langle k_{\text{in}}^2 \rangle_h \approx 0.36$ is much larger than the second moment of a Poisson distribution with the same average degree $\langle k^2 \rangle_{\text{ER}} \approx 0.28$.

By choosing one day time steps we assume slower dynamics on the network. The coarse graining removes the daily activity patterns. To study control spanning over multiple weeks, we explicitly remove weekends and holidays, i.e. we measure the time in workdays. The average

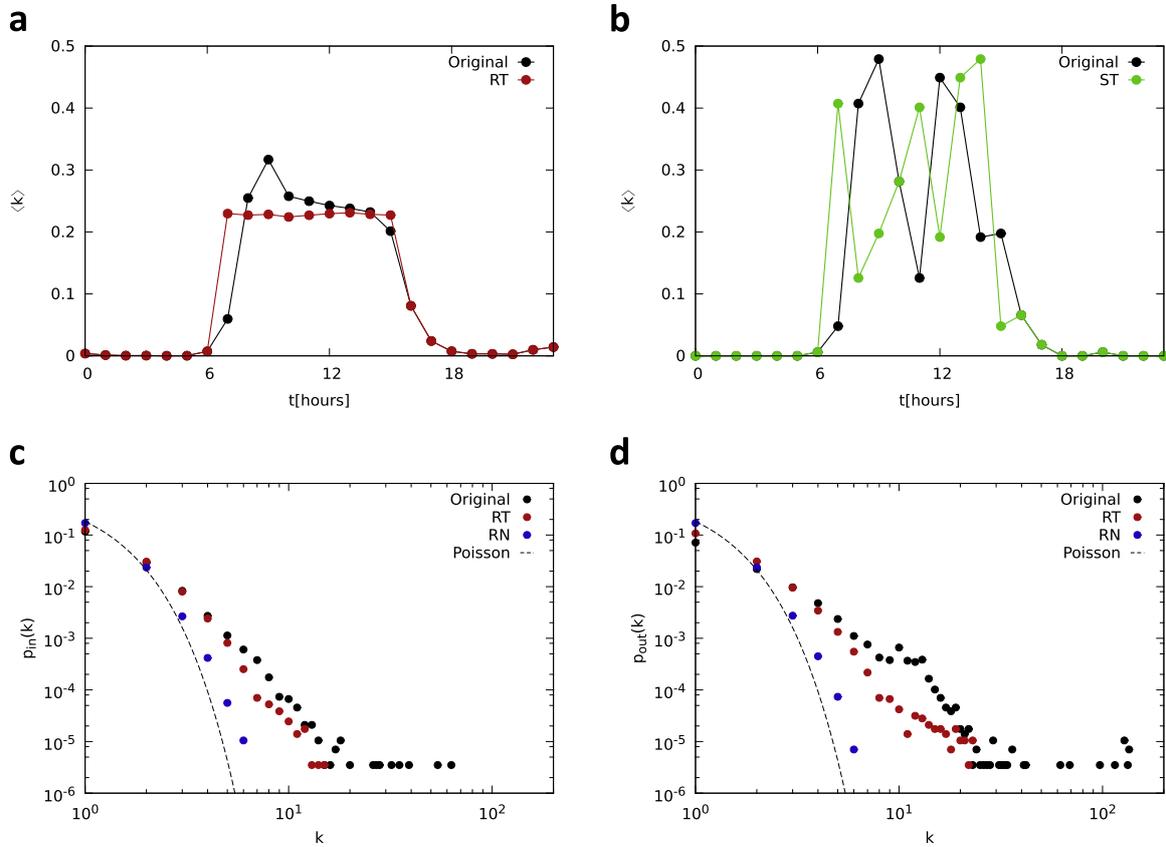


Figure C1. The temporal network of email communication with one hour time resolution. (a) The average degree of the network at different hours of the day; the plot shows the average of all workdays. The bulk of the email traffic happens during a 9 hour period corresponding to office hours (black). We eliminate the fluctuations in the average degree by assigning random times to the links within the 9 hour active period (RT, red). (b) The average degree of the network at different hours of a typical workday. The average degree is characterized by large fluctuations. Shuffling the time steps keeps the fluctuations, but eliminates correlations between consecutive layers (ST, green). (c) and (d) The in- and out-degree distribution within a time step, the plot shows the average of all working hours. The original distributions are highly heterogeneous (black) compared to the Poisson distribution with the same average. Assigning random link times indirectly changes the distributions; however, the distributions remain heterogeneous (RT, red). Randomizing the network topology within a time step eliminates the heterogeneity (RN, blue). Note that the RN randomization keeps the fluctuations in the average degree, hence the difference from the Poisson distribution. To obtain the Poisson distribution, we have to apply both the RT and the RN randomization. DPN and ST randomization do not change the degree distribution.

degree within a time step is $\langle k \rangle_d \approx 1.76$ (figure C2(a)), which predicts that the system is in the supercritical phase, meaning that the characteristic control time is in the order of the system size. Therefore, the length of the available time period does not allow multiple independent measurements of the control process, hence we focus on controlling the system at the end of the last workday at $t = 190$. Similarly to the one hour case, the average degree distribution within a time step is heterogeneous (figures C2(b) and (c)), with second moments $\langle k_{out}^2 \rangle_d \approx 18.07$ and

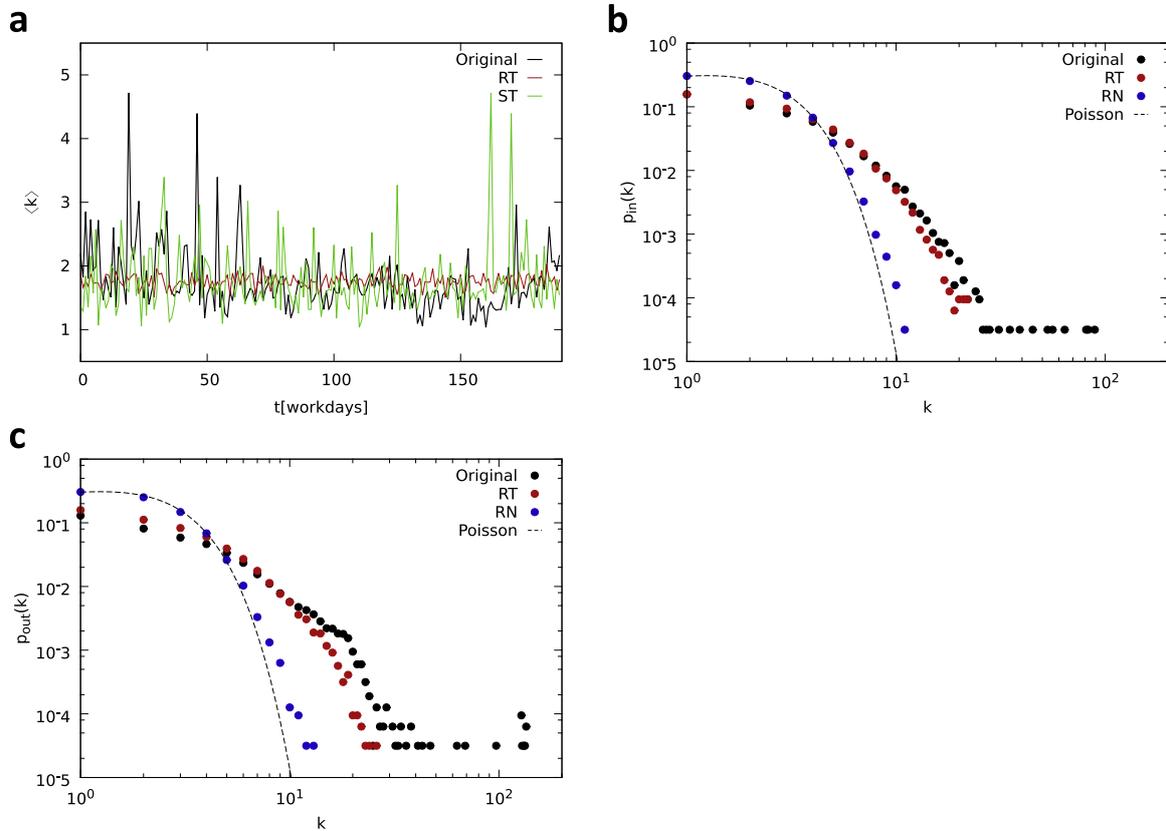


Figure C2. The temporal network of email communication with one day time resolution. (a) The average degree of the network at different days. Similarly to the one hour case, the average degree has large fluctuations. Randomly assigning link times eliminates the fluctuations (RT, red), and shuffling the time steps only eliminates correlations between consecutive time steps (ST, green). (b) and (c) The in- and out-degree distribution within a time step. The original distributions are highly heterogeneous (black) compared to the Poisson distribution with the same average. Assigning random link times indirectly changes the distributions, however, the distributions remain heterogeneous (RT, red). Randomizing the network topology within a time step eliminates the heterogeneity (RN, blue). DPN and ST randomization do not change the degree distribution.

$\langle k_{in}^2 \rangle_d \approx 11.12$) compared to the second moment assuming a Poisson distribution with the same average degree $\langle k^2 \rangle_{ER} \approx 4.86$.

C.2. Randomization procedures

We use four different randomization techniques to identify which temporal or network characteristics of the system influence controllability. For details on randomizing temporal networks see [18] and references within.

Random time (RT): this randomization assigns random time steps to each link, thereby removing all temporal correlations, both overall fluctuations in the average degree, and local correlations such as consequent and simultaneous events (figures C1(a) and C2(a)). This randomization does not change who interacts with whom, that is, it does not change the

aggregated network. However, by separating simultaneous events, the randomization changes the degree distribution within a time step indirectly (figures C1(c) and (d) and C2(b) and (c)). For the one hour coarse grained network, we only randomize within the working hours of each workday.

Shuffled time (ST): we shuffle the order of all of the time steps. This removes all correlations between subsequent time steps, such as casual chain of events, while the structure within a time steps remains unchanged (figures C1(b) and C2(a)). For the one hour coarse grained network, we only shuffle the time steps within the working hours of each workday.

Random network (RN): in this randomization, the network for each time step is replaced by an Erdős–Rényi network with the same number of links, thereby removing all network structure, including the heterogeneity from the degree distribution (figures C1(c) and (d) and C2(b) and (c)). All interaction times are retained, preserving the fluctuations in the average degree.

Degree preserved network (DPN): for this randomization, we break all connections, and randomly rewire them within a time step. This way only the degree distribution is preserved, but all other correlations in the network structure are eliminated. Similarly to RN, we do not change the interaction times.

References

- [1] Albert R and Barabási A-L 2002 Statistical mechanics of complex networks *Rev. Mod. Phys.* **74** 47–97
- [2] Newman M E J 2003 The structure and function of complex networks *SIAM Rev.* **45** 167–256
- [3] Barrat A, Barthélemy M and Vespignani A 2008 *Dynamical Processes on Complex Networks* vol 135 (Cambridge: Cambridge University Press)
- [4] Fortunato S 2010 Community detection in graphs *Phys. Rep.* **486** 75–174
- [5] Wang X F and Chen G 2002 Pinning control of scale-free dynamical networks *Physica A* **310** 521–31
- [6] Lombardi A and Hörnquist M 2007 Controllability analysis of networks *Phys. Rev. E* **75** 056110
- [7] Fiedler B, Mochizuki A, Kurosawa G and Saito D 2013 Dynamics and control at feedback vertex sets i. informative and determining nodes in regulatory networks *J. Dyn. Differ. Equ.* **25** 563–604
- [8] Cornelius S P, Kath W L and Motter A E 2013 Realistic control of network dynamics *Nat. Commun.* **4** 1942
- [9] Liu Y-Y, Slotine J-J and Barabási A-L 2011 Controllability of complex networks *Nature* **473** 167–73
- [10] Liu C T 1974 Structural controllability *IEEE Trans. Autom. Control* **19** 201–8
- [11] Hosoe S 1980 Determination of generic dimensions of controllable subspaces and its application *IEEE Trans. Autom. Control* **25** 1192–6
- [12] Nepusz T and Vicsek T 2012 Controlling edge dynamics in complex networks *Nat. Phys.* **8** 568
- [13] Yan G, Ren J, Lai Y-C, Lai C-H and Li B 2012 Controlling complex networks: how much energy is needed? *Phys. Rev. Lett.* **108** 218703
- [14] Pósfai M, Liu Y-Y, Slotine J-J and Barabási A-L 2013 Effect of correlations on network controllability *Sci. Rep.* **3** 1067
- [15] Jia T *et al* 2013 Emergence of bimodality in controlling complex networks *Nat. Commun.* **4** 2002
- [16] Sun J and Motter A E 2013 Controllability transition and nonlocality in network control *Phys. Rev. Lett.* **110** 208701
- [17] Yuan Z, Zhao C, Di Z, Wang W-X and Lai Y-C 2013 Exact controllability of complex networks *Nat. Commun.* **4** 2447
- [18] Holme P and Saramäki J 2012 Temporal networks *Phys. Rep.* **519** 97–125
- [19] Barabási A-L 2005 The origin of bursts and heavy tails in human dynamics *Nature* **435** 207

- [20] Malmgren R D, Stouffer D B, Motter A E and Amaral L A 2008 A poissonian explanation for heavy tails in e-mail communication *Proc. Natl Acad. Sci.* **105** 18153–8
- [21] Vazquez A, Rácz B, Lukács A and Barabási A-L 2007 Impact of non-poissonian activity patterns on spreading processes *Phys. Rev. Lett.* **98** 158702
- [22] Iribarren J L and Moro E 2009 Impact of human activity patterns on the dynamics of information diffusion *Phys. Rev. Lett.* **103** 038702
- [23] Jo H-H, Karsai M, Kertész J and Kaski K 2012 Circadian pattern and burstiness in mobile phone communication *New J. Phys.* **14** 013055
- [24] Pan R K and Saramäki J 2011 Path lengths, correlations, and centrality in temporal networks *Phys. Rev. E* **84** 016105
- [25] Korschake M, Lentz H H K, Conraths F J, Hövel P and Selhorst T 2013 On the robustness of in- and out-components in a temporal graph *PLoS One* **8** e55223
- [26] Kwakernaak H and Sivan R 1972 *Linear Optimal Control Systems* (New York: Wiley)
- [27] Hartung C, Reissig G and Svaricek F 2013 Sufficient conditions for strong structural controllability of uncertain linear time-varying systems *American Control Conf. (ACC)* **2013** 5875–80
- [28] Harris T E 1963 *The Theory of Branching Processes* (Berlin: Springer)
- [29] Liu Y-Y, Csóka E, Zhou H and Pósfai M 2013 Core percolation on complex networks *Phys. Rev. Lett.* **109** 205703
- [30] Zdeborová L and Mézard M 2006 The number of matchings in random graphs *J. Stat. Mech.* P05003
- [31] Friedkin N E 1986 A formal theory of social power *J. Math. Sociol.* **12** 103–26
- [32] Tanner H G 2004 On the controllability of nearest neighbor interconnections *CDC 43rd IEEE Conf. on Decision and Control* pp 2467–72
- [33] Michalski R, Palus S and Kazienko P 2011 Matching organizational structure and social network extracted from email communication (*Lecture Notes in Business Information Processing* vol 87) (Berlin: Springer) pp 197–206
- [34] Cowan N J, Chastain E J, Vilhena D A, Freudenberg J S and Bergstrom C T 2012 Nodal dynamics, not degree distributions, determine the structural controllability of complex networks *PLoS One* **7** e38398
- [35] Zhao C, Wang W-X, Liu Y-Y and Slotine J-J 2014 Universal symmetry in complex network control in press (arXiv:1403.0041)
- [36] Lai Y-C 2014 Controlling complex, non-linear dynamical networks *Natl Sci. Rev.* **1** 339–41
- [37] Liu Y-Y, Slotine J-J and Barabási A-L 2012 Control centrality and hierarchical structure in complex networks *PLoS One* **7** e44459
- [38] Ford L R and Fulkerson D R 1962 *Flows in Networks* (Princeton: Princeton University Press)
- [39] Newman M E J 2010 *Networks: An Introduction* (New York: Oxford University Press)
- [40] Caldarelli G, Capocci A, De Los Rios P and Muñoz M A 2002 Scale-free networks from varying vertex intrinsic fitness *Phys. Rev. Lett.* **89** 258702
- [41] Söderberg B 2002 General formalism for inhomogeneous random graphs *Phys. Rev. E* **66** 066121
- [42] Erdős P and Rényi A 1959 On random graphs *Publ. Math. Debrecen* **6** 290–7
- [43] Barabási A L and Albert R 1999 Emergence of scaling in random networks *Science* **286** 509
- [44] Goh K-I, Kahng B and Kim D 2001 Universal behavior of load distribution in scale-free networks *Phys. Rev. Lett.* **87** 278701
- [45] Harris T E 1963 *The Theory of Branching Processes* (Berlin: Springer)
- [46] Karp R and Sipser M 1981 Maximum matchings in sparse random graphs *Proc. 22nd FOCS* pp 364–72
- [47] Bauer M and Golinelli O 2001 Core percolation in random graphs: a critical phenomena analysis *Eur. Phys. J. B* **24** 339–52
- [48] Michalski R, Palus S and Kazienko P 2011 Matching organizational structure and social network extracted from email communication (*Lecture Notes in Business Information Processing* vol 87) (Berlin: Springer) pp 197–206
- [49] Konect network dataset - konect (2013) <http://konect.uni-koblenz.de/networks/konect>