AFRL-RI-RS-TR-2014-021

ANALYSIS AND DESIGN OF COMPLEX NETWORK ENVIRONMENTS

BRIGHAM YOUNG UNIVERSITY

FEBRUARY 2014

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2014-021 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /
ROBERT KAMINSKI
Work Unit Manager

/ S /
WARREN H. DEBANY JR.
Technical Advisor, Information
Exploitation and Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEB 2014	FINAL TECHNICAL REPORT	AUG 2011 – AUG 2013

4. TITLE AND SUBTITLE

ANALYSIS AND DESIGN OF COMPLEX NETWORK ENVIRONMENTS

5a. CONTRACT NUMBER
FA8750-11-1-0236

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

SEAN WARNICK, DANIEL ZAPPALA

5d. PROJECT NUMBER
UGOV

5e. TASK NUMBER
BY

5f. WORK UNIT NUMBER
U2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Brigham Young University
Office of Research & Creative Activities, Department of Computer Science
University Hill
Provo, UT 84602-1231

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIG
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2014-021

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This work extends previous research employing 1) dynamical structure functions as a new modeling tool for complex network environments, and 2) the network design cycle as a mechanism for unifying theoretical and applied research. In particular, this work explores a specific application to a wireless mesh network environment and extends the theoretical results to include necessary and sufficient informativity conditions for network reconstruction from input-output data, vulnerability analyses of both open-loop and closed-loop architectures, and the synthesis of distributed feedback controllers with arbitrary network structure constraints.

15. SUBJECT TERMS

Complex Networks, Network Design, Wireless Mesh Networks, Bio-chemical Reaction Networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**ROBERT L. KAMINSKI**
U	U	U	SAR	48	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

Contents

Figures

List of Figures

Tables

List of Tables

1.0 Summary

This work presents results exploring the use of dynamical structure functions, and their associated signal structure, as a modeling tool to map and detect the interference structure in wireless mesh networks. As wireless networks become more pervasive, it is increasingly important to have efficient algorithms for mapping the network and detecting interfering devices that can impact performance. Recent work on partial structure representations of linear time invariant controlled dynamic systems offers a vehicle for addressing these issues in new and efficient ways. In particular, we use the signal structure, which is the open-loop causal relationships among signals, to represent causal relationships between links in a wireless network. To accomplish this, we develop a smooth, differential equation model for the 802.11 MAC and validate it using ns-3 simulations. We then propose a network mapping protocol that constructs the signal structure of the network during live operation using $O(n)$ perturbation experiments. Each experiment deliberately changes the sending rate on a particular link, while monitoring the actual rates realized across the network. The resulting map indicates causal relationships between link rates, rather than its physical topology. This allows us to detect intrusive devices that temporarily disrupt communication and impair system performance.

To support this use of the signal structure for mapping interference in wireless mesh networks, we also extend previous theoretical work in the conditions needed to reconstruct a network from data. In particular, we demonstrate precisely the a priori structural information that is both necessary and sufficient to reconstruct a network from input-output data.

We then consider the security issues arising from networks with particular structures and develop a theory of vulnerability for both open and closed loop systems. We consider destabilizing attacks, which are attacks on a single link within a system that could potentially destabilize the entire system. We define the vulnerability of a link to be the inverse of the effort required on that link to destabilize the system. With these definitions, we formulate two problems where we wish to minimize vulnerability. The first, the open-loop problem, exists when we are given a fixed system design and have complete control over its implementation. We show that for such a system, we can always create a completely-secure implementation an implementation with zero vulnerability if we eliminate all internal feedback from the systems structure. The second, the closed-loop problem, occurs when there exists some restrictions to the systems implementation. In particular, we consider the case where the system is implemented by two subsystems of given design are connected in feedback. We show that the vulnerability of one system is only dependent on the design and not the implementation of the other. We then show that removing internal feedback from a subsystem does not necessarily minimize vulnerability; that it is possible to fight fire with fire and use internal feedback to combat the vulnerability introduced by connecting systems in feedback.

Finally, we consider the problem of synthesizing systems with a particular network structure. In particular, we consider the design of feedback controllers that are designed to stabilize a given unstable system but must have a particular (specified) network structure. Although this problem is known to be hard in general, we consider a particular heuristic and demonstrate that either our design process yields stabilizing controller with the desired structure, or no such controller exists.

2.0 Introduction

The study of network environments have received considerable attention from various disciplines over the last 20 years. This is in part due to the rise of the internet, but it also is due to advances in biology that have brought new kinds of networked "machines" into clear view that contrast sharply with our solid-state, engineered systems. The research discussed here takes a fresh view of these new, "complex" network environments and answers fundamental questions about 1) the design of experiments necessary to discover their structure (and thus adapt system inputs to optimize the resulting performance), and 2) the relation- ship between network structure to vulnerability and attack, and 3) the design of stabilizing feedback systems that respect a particular network structure.

Specifically, this work explores these issues in the context of both wireless mesh networks. This research unites theoretical work that clarifies fundamental limitations of complex networks with network engineering to implement specific designs and experimentally verify the theoretical discoveries in what we call the "network design cycle." Although more work is needed to compare that behavior of real networks with our models, this work lays a clear foundation for subsequent results.

2.1 What is a "complex" networked environment?

The descriptor "complex" has been used in other studies to characterize networks that are large, meaning that a graph used to represent the network has a large number of nodes. Often the number of edges is also taken to be large and devoid of certain regular structure, so that, for example, a tree-structured graph would not be considered "complex." In these studies one typically looks for hidden regularities or patterns that characterize the structure of the graph in simple terms in spite of these other "complexities." A typical example would be small-world networks.

In this work, we consider different environments, where "network" refers to an interconnected dynamical system. In these environments, system behavior is as important a characteristic of the network as is its structure. The descriptor "complex" then refers to both the network behavior as well as its structure.

This work addresses networks with complex behavior by considering systems with underlying nonlinear dynamics and noisy measurements. As a first step, the results presented here leverage Lyapunov theory to apply linear analysis locally to regions near equilibria of the underlying nonlinear system. Extending these results globally to the underlying non- linear system is not immediately obvious and requires new thinking about the meaning of structure, beyond that already developed in this research. As a result, global analysis is left for future research.

This work also addresses networks with complex structure. Besides the usual definition, of a "complex" network structure represented by a graph with a large number of nodes and non-trivial edge patterns (e.g. allowing for arbitrarily complicated feedback relationships), this work also makes a particular contribution by developing representation and analysis tools applicable to networks of dynamical systems with potential hidden entanglements among unmeasured variables. This "potential entanglement" type of network complexity is previously unaddressed in the literature, yet it becomes particularly important for inferring network

structure from behavioral data.

Appreciating the power of structural representations that allow for potential entanglement among unmeasured variables to simplify network inference problems is subtle, but it is a central contribution of this research program. Consider, for example, a network that is composed of the interconnection of various subsystems. By definition, each subsystem's internal states only affect that subsystem, and the interconnection variables are themselves measured quantities. Inferring this network "subsystem" interconnection structure from data thus demands the discovery of the true partition of all of the system's *unmeasured* states into their appropriate subsystem components. Discovering such a partition from measured data can be extremely difficult or impossible. This research, on the other hand, leverages a different representation of system structure, called signal structure, that does not rely on the idea of subsystems and allows for potential entanglement among unmeasured states. As a result, inferring a system's signal structure requires much less information, and thus fewer experiments, than inferring a system's subsystem structure.

One contribution of this research is to thoroughly understand the relationships between a system's subsystem and signal structures. Often, systems with solid-state components (such as routers in the internet) have subsystem and signal structures that are equivalent. Sometimes, however, systems have a fluid-like character (such as bio-chemical reaction networks or wireless mesh networks) and the resulting subsystem and signal structures can be very different. Characterizing informativity conditions and developing scalable algorithms for network reconstruction (i.e. inference) of signal structure are among the primary theoretical contributions of this effort. These models of complex networked environments also facilitate a novel robustness analysis that leads to new results about system vulnerability and security. Moreover, designing systems that meet network constraints yet accomplish specific tasks, such as the stabilization of a given system, is known to be a hard problem. Nevertheless, we offer a heuristic that either results in the desired design or proves no such design exists. These contributions are highlighted when applied to complex network environments that exhibit both the behavioral and structural complexity as described.

2.2 What is the network design cycle?

Besides the theoretical contributions of this work, this research represents an active collaboration between theoretical development and physical implementation and testing on real complex networks. This collaboration is most evident in our work on wireless mesh networks, where active modeling and protocol design efforts lead to simulation, implementation, and experimental testing on our live wireless mesh test-bed. Similar collaborations for bio-chemical reaction networks began to emerge during this study, but the implementation and experimental testing is incomplete and part of ongoing research.

The network design cycle defines the scientific process we engage that unites our theoretical and applied work. The next section discusses how we use the Network Design Cycle to interconnect our theoretical results to applications, such as wireless mesh networks. Section 3 then details our results modeling wireless mesh networks, developing informatively conditions for network reconstruction, introducing a theory of vulnerability, and presenting a heuristic for designing stabilizing distributed feedback controllers with a specific network structure (or showing that no such controller exists).

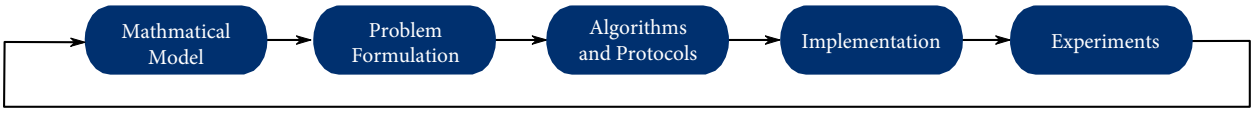


Figure 1: The Network Design Cycle

3.0 Methods, Assumptions, and Procedures

To address these problems, our work uses a method we call the Network Design Cycle, as shown in Figure 1. We start by formulating a mathematical model of the network, precisely characterizing how it operates. In a wireless network, this may involve describing how nodes interact via carrier sensing and interference. Based on this model, we carefully formulate the research problem, such as providing optimal performance or detecting malicious nodes. The next step is to design algorithms and protocols that will solve the problem. We then implement the solution, which often requires considering additional complexities that arise when building and deploying code on a network. Finally, we run experiments to validate that the solution works as designed, for example by providing optimal performance or security guarantees. If performance deviates from what is expected, this means that our original model or problem formulation was wrong. This leads to another iteration of the design cycle, where we create a refined model or reformulate the problem.

For the steps that involve implementation and experimental evaluation, we use a variety of tools. We use MATLAB to numerically evaluate algorithms, ensuring that they converge to the expected solution and, in some cases, provide optimal performance. For experimental testing, we use a wireless mesh network deployed in our department's building, consisting of 30+ computers configured as wireless routers, spread over two floors. The mesh network can be configured so that it uses 802.11a, and we ensure that we select a channel with no other traffic, so that we can run experiments without any outside noise when desired. Our protocols are implemented using WiFi, a toolkit developed with funding from NSF that enables rapid deployment of experimental wireless protocols in user-space.

4 Results and Discussion

4.1 Using signal structure to map and detect interference in wireless mesh networks

4.1.1 Background

Dynamical structure functions are a representation for linear time invariant systems developed in [5]. A DSF gives a partial representation of the structure of the system, namely how the inputs affect the manifest states and how the manifest states affect each other. A brief derivation is provided below; for a full derivation see [17].

Let us consider a state-space LTI system

$$\begin{bmatrix} \dot{y} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \qquad (1)$$
$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix},$$

Here y are the states that are measured, and x are the hidden states. Note that the assumption in the second equation is made for notational convenience.

Taking Laplace Transforms of the signals in (1), we get

$$\begin{bmatrix} sY \\ sX \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U. \qquad (2)$$

Solving for X in the second equation of 2 gives

$$X = (sI - A_{22})^{-1} A_{21} Y + (sI - A_{22})^{-1} B_2 U$$

Substituting into the first equation of (2) we get,

$$sY = WY + VU,$$

where $W = A_{11} + A_{12}(sI - A_{22})^{-1} A_{21}$ and $V = A_{12}(sI - A_{22})^{-1} B_2 + B_1$. Let D be a diagonal matrix with the diagonal entries of W. Then,

$$(sI - D)Y = (W - D)Y + VU.$$

Now we can rewrite this equation as,

$$Y = QY + PU, \qquad (3)$$

where

$$Q = (sI - D)^{-1}(W - D)$$

$$P = (sI - D)^{-1} V.$$

The matrix Q is a matrix of transfer functions from Y_i to Y_j, $i \neq j$, relating each measured signal to the other measured signals. A nonzero entry in Q_{ji} says that the signal Y_i affects the signal Y_j either directly or through some hidden states. Note that Q is zero on the diagonal and either zero or a strictly proper transfer function on the off diagonal. The matrix P is a matrix of zeros or strictly proper transfer functions from each input to each output without depending on any additional measured states. Together, the pair $(Q(s), P(s))$ is called the *dynamical structure function* for system (1). The transfer function matrix for this system is given by

$$G = (I - Q)^{-1} P = C(sI - A)^{-1} B.$$

An example of the dynamical structure function representation of an LTI system is given in

Example1. *Example 1. Let us consider a system with two measured states given by the following state space equation.*

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -4 & 0 & 1 \\ 0 & -3 & 2 \\ 3 & 2 & -3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ x_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ x_1 \end{bmatrix},$$

The DSF for this system is given by,

(4)

$$Q = \begin{bmatrix} 0 & \frac{2}{s^2+7s+9} \\ \frac{6}{(s+1)(s+5)} & 0 \end{bmatrix} \text{ and}$$

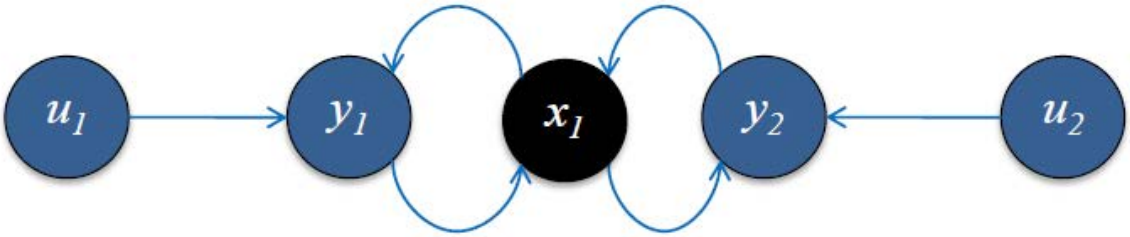
$$P = \begin{bmatrix} \frac{s+3}{s^2+7s+9} & 0 \\ 0 & \frac{1}{2(s+1)} + \frac{1}{2(s+5)} \end{bmatrix},$$

and the transfer function is given by

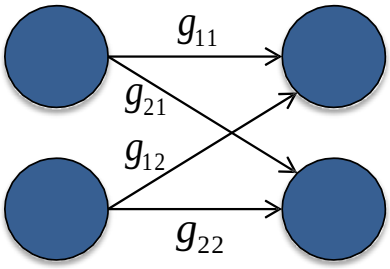
$$G = \begin{bmatrix} \frac{(s+1)(s+5)}{s^3+10s^2+26s+11} & \frac{2}{s^3+10s^2+26s+11} \\ \frac{6}{s^3+10s^2+26s+11} & \frac{s^2+7s+9}{s^3+10s^2+26s+11} \end{bmatrix}.$$

Figure 2 shows a graphical view of this system in various representations. Figure 2(a) shows the state space realization of the system, which contains information about the dependency among input, state, and output variables. Essentially, the state space of the system defines both the structure and dynamics of the entire network. A simpler representation is shown in Figure 2(b), the system's transfer function contains the dynamics of the system, but yields no information about the structure of the network. In Figure 2(c) the DSF of this system shows the relationship between the measured states, y_1 and y_2, something not visible from the system's transfer function. In the situations when a complete state space model of the system cannot be obtained, a DSF model of the system can be used to obtain a partial structure of the system. Note that when all the states are measured, DSF models give the actual structure of the system.

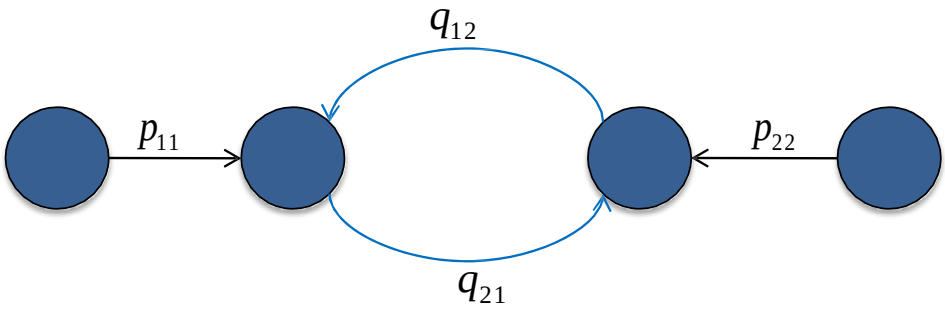
For a given network system, if we know sufficient information about how the inputs affect the measured states then we can reconstruct the DSF of the system from the input-output data. A sufficient condition for the reconstructibility is that if *P* is square and diagonal and *G* is invertible[5]. This method has been used successfully to identify the structure of biochemical networks in [7]. Since these biological systems are highly nonlinear, an equilibrium is chosen for the reconstruction. Then, the input is provided as small perturbations to the equilibrium input and the corresponding perturbed outputs are measured. Once the inputs and equilibrium outputs are collected, reconstruction can be done using the technique described in [18].



(a) Complete structure of the system (4). This can be obtained if the state space model of the system is known.



(b) Structure given by the Transfer Function of system (4).



(c) Structure given by the DSF of system (4).

Figure 2: Structures given by: (a) the state space, (b) the transfer function, and (c) the dynamical structure function. Here, x_1 is a hidden state: it is not measured directly and as a result is not seen in the DSF representation.

4.1.2 Modeling the 802.11 MAC

Our mapping methodology uses the reconstruction method given in [18], which assumes that the actual nonlinear system is smooth, i.e. it can be modeled as a system of differential equations. However, we know that wireless networks are discrete event systems with dynamics determined by the choice of the sending nodes to send packets at certain times. In this section, we show that although wireless networks are discrete event systems, their equilibrium properties as well as the transition dynamics can be captured by differential

equations. **Model Description**

We seek to develop a continuous, differential model for the carrier sensing and back-off behavior of the 802.11 MAC. Packets sent to the MAC layer from TCP and IP are queued in the MAC layer until they can be transmitted. Queued packets are sent as soon as possible, leading the MAC to consume all available bandwidth. Instantaneous sending rates are either zero or the full link speed. To facilitate analysis, we consider instead the average rate over a sliding window. This smooths the discontinuities of instantaneous rates and allows us to accurately model rate dynamics with differential equations.

We consider wireless networks from a link point of view. Let x_i denote the normalized sending rate of link i and let b_i denote the buffer size. The rate of change in the buffer size is the difference between the rate of packet arrival and departure from the buffer, i.e. $\frac{db_i}{dt} = u_i - x_i$ where u_i is the rate at which packets are sent from higher layers to the MAC layer.

The dynamics for x_i should move x_i to consume the available rate. To determine how much bandwidth is available to link x_i, it is useful to view normalized rates as the probability that a particular link is active at a given time. The bandwidth available to link i is the probability that none of the links are active which contend with link i.

Let X_i denote a Bernoulli random variable indicating whether link i is broadcasting, i.e., $P(X_i = 1) = x_i$ and $P(X_i = 0) = 1 - x_i$. Similarly, let $\bar{X}_i$ be a Bernoulli random variable indicating whether any link contending with link i is broadcasting. By the Law of Total Probability, the bandwidth available to link i is

$$\begin{aligned} P(\bar{X}_i = 0) =& P(X_i = 1)P(\bar{X}_i = 0 \mid X_i = 1) \\ &+ P(X_i = 0)P(\bar{X}_i = 0 \mid X_i = 0). \end{aligned} \tag{5}$$

We assume that two contending links never broadcast concurrently, so $P(\bar{X}_i = 0 \mid X_i = 1) = 1$. We assume temporarily, for clarity of exposition, that link i is the only common link between cliques.

To avoid wasting bandwidth, the available rate should only influence the dynamics for link i when there are buffered packets ready to be sent. This effect could be modeled using a step function but would introduce a discontinuity into the dynamics. Instead, we use a sigmoid as a continuous approximation, i.e., $\sigma(b) = \frac{1}{1+e^{-\alpha b}}$ where α dictates the slope of the sigmoid at zero. For our simulations, we let $\alpha = 1$.

The 802.11 MAC dictates short periods of mandatory silence following each transmission. In addition to fixed length delays, stations are required to wait for a period of random duration before transmitting. This is done to avoid collisions caused by multiple stations attempting to broadcast immediately following the end of a transmission. The random back-off time is chosen uniformly within a range known as the contention window. The combination of mandatory silence and the random back-off decreases the maximum achievable throughput. We denote the proportion of achievable throughput for link i by $\beta_i \in (0, 1)$.

The overall dynamics for a topology having at most one common link between any two cliques are

$$\dot{b}_i = u_i - x_i \tag{6}$$
$$\dot{x}_i = -x_i + \beta_i \sigma(b_i) \left(x_i + (1 - x_i) \prod_{j \in C_i} \left(1 - \frac{\sum_{k \in L_j \setminus i} x_k}{1 - x_i} \right) \right)$$

To generalize the result to allow for more overlap between cliques, let

$$O_i = \{i\} \cup \{j : \exists k, l,\ k \neq l,\ j \in L_k,\ j \in L_l, \text{ and } k, l \in C_i\}$$

denote the set of links common to multiple cliques in C_i. The union with $\{i\}$ ensures the model simplifies correctly in the case where link i belongs to a single clique. Allowing for this in the preceding development leads to the dynamics

$$\dot{b}_i = u_i - x_i \tag{7}$$
$$\dot{x}_i = -x_i + \beta_i \sigma(b_i) \Bigg(x_i + \prod_{j \in O_i} (1 - x_j) \prod_{j \in C_i} \left(1 - \frac{\sum_{k \in L_j \setminus O_i} x_k}{1 - \sum_{k \in O_i \cap L_j} x_k} \right) \Bigg).$$

Methodology

To validate our model we compare a MATLAB computation of the differential model to packet-level ns-3 simulations. (For validations of the ns-3 simulator itself, see [3].) We simulate various contention relationships and monitor all carrier sensing behavior in the network. We calculate rates for a link as the normalized ratio of successful channel accesses as compare to total airtime.

For each simulation, we create a wireless network consisting of a set of senders and receivers placed a fixed distance from one another. An application at each sender transmits packets of uniform size to its designated receiver counterpart at a specified rate. The u parameter of our model represents the rate at which packets are sent from higher layers

to the MAC layer. Host start times are staggered in one second increments, and run for a duration of 500 seconds. During this time, whenever a host successfully captures the medium, i.e. clears carrier sensing, a logger hooked into the MAC layer records this information along with a timestamp. It is the collection of this data across all sending hosts in the network that comprises our data set.

Each host in the simulated environment houses a full network stack, utilizing the IEEE 802.11g protocol for communication. Link speeds are set using a baseline of 6 Mbps, the default 802.11 access point broadcast rate. End-to-end transmission is achieved via the UDP transport protocol, avoiding the extra complexities of reliability and congestion control inherent in TCP. Standard Ethernet MTU packet sizes of 1500 bytes are used, and MAC layer retries are disabled, so as to avoid influencing simulation results.

Validation

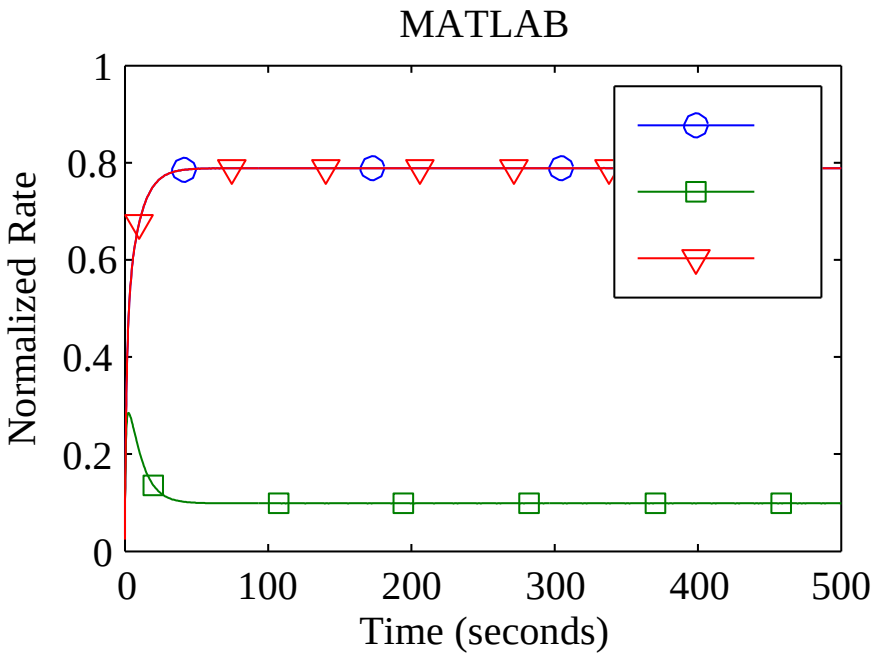
We validate our model using three different topologies, involving paths and cycles of various lengths. We construct chain topologies such that all nodes are equally spaced along a straight line, and for any three sequential nodes along the path, the left node and right node are in contention range only of the center node and not with one another, whereas the center node is in contention range with both edge nodes. We construct cycle topologies as regular n-sided polygons, where each node contends with exactly two other nodes, its direct neighbors.

We begin the validation with a 3-chain, or a man-in-the-middle topology. As shown in Fig. 3, both our model and the simulator accurately capture the starvation of the link in the middle, a well-known result. In this case, our model uses $U = [1\ 1\ 1]$, $\alpha = 1$, and $\beta = 0.875$. A controller for this network would limit the sending rates of the outer links, allowing the link in the middle more bandwidth. When the outer links are given reduced rates, our model agrees very closely with ns-3 in giving the additional bandwidth to the link in the middle. To do this, the β parameter in our model must be varied, a direct result of differing contention window sizes affecting how much of the channel capacity is lost to overhead.

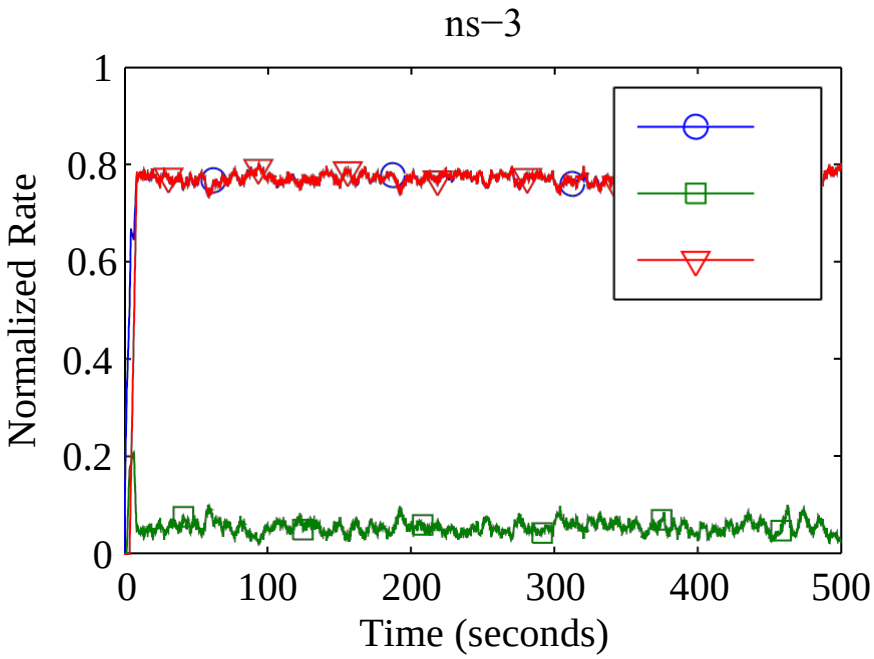
The next topology we consider is a chain of six links. As shown in Fig. 4, both our model and ns-3 capture the behavior of the MAC properly in this case. In this case our model uses $U = [1\ 1\ 1\ 1\ 1\ 1]$, $\alpha = 1$, and $\beta = 0.85$. Removing the contention between links 3 and 4 would result in two man-in-the-middle topologies starving links 2 and 5. However, with this contention between links 3 and 4 reduces their achievable sending rates, which in turn allows more airtime for links 2 and 5.

Fig. 5 shows simulation results from a cycle of six links. In this case our model uses $U = [1\ 1\ 1\ 1\ 1\ 1]$, $\alpha = 1$, and $\beta = 0.85$. The ns-3 results for this topology show considerable short-term variation in link rates over time. This is a result of short-term (on the order of a second) starvation among the links. Stations alternate between starvation and exclusively capturing the medium within the link's cliques. The irregular alternations combined with a sliding window average results in the plot shown in Fig. 5. We reiterate that our model does not attempt to predict such fast dynamics but seeks to accurately predict average rates.

Finally, we use a bowtie topology, consisting of a four links arranged in a square, with a single link in the middle. Like the man-in-the-middle topology, the middle link is easily starved. The differential model is again accurate in capturing the dynamics and steady-state behavior of the MAC in this case, with $U = [1\ 1\ 1\ 1\ 1]$, $\alpha = 1$, and $\beta = 0.9$.

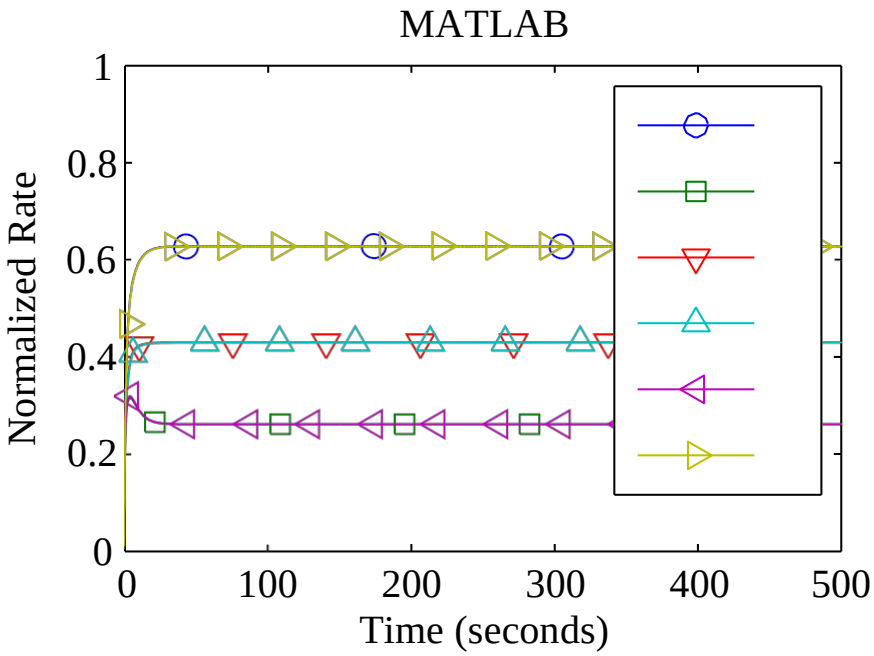


(a) differential model

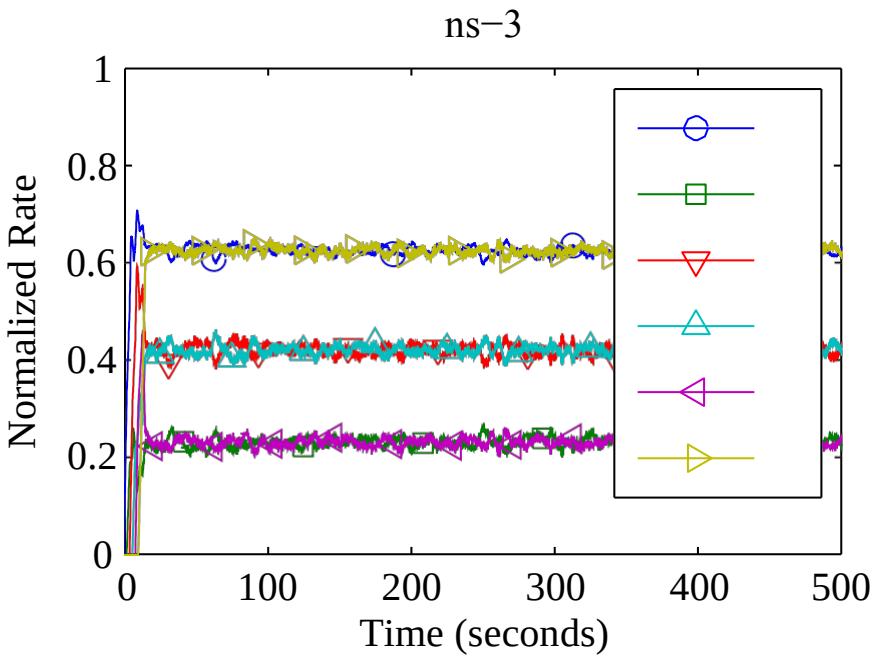


(b) ns-3

Figure 3: Model validation, man-in-the-middle graph

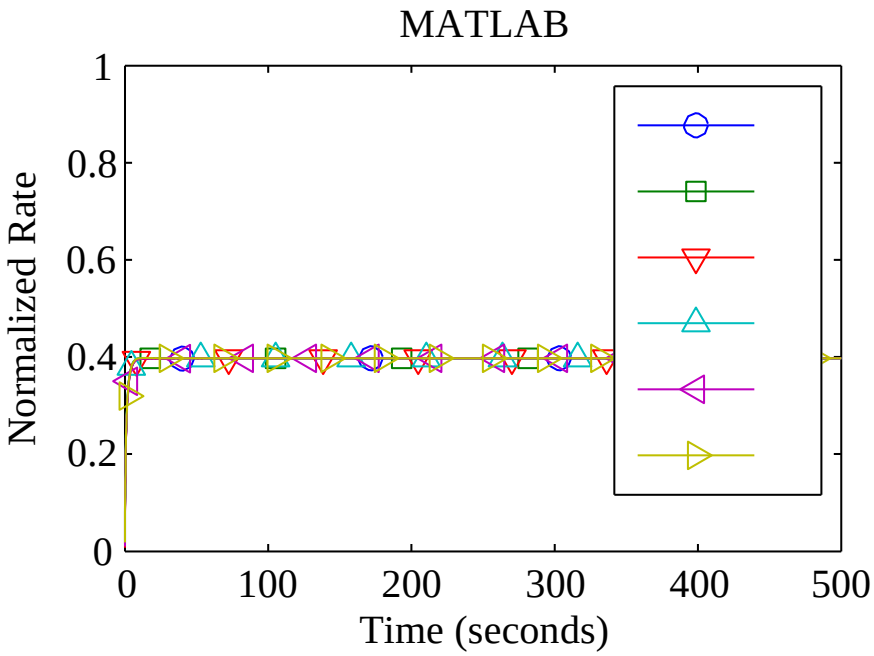


(a) differential model

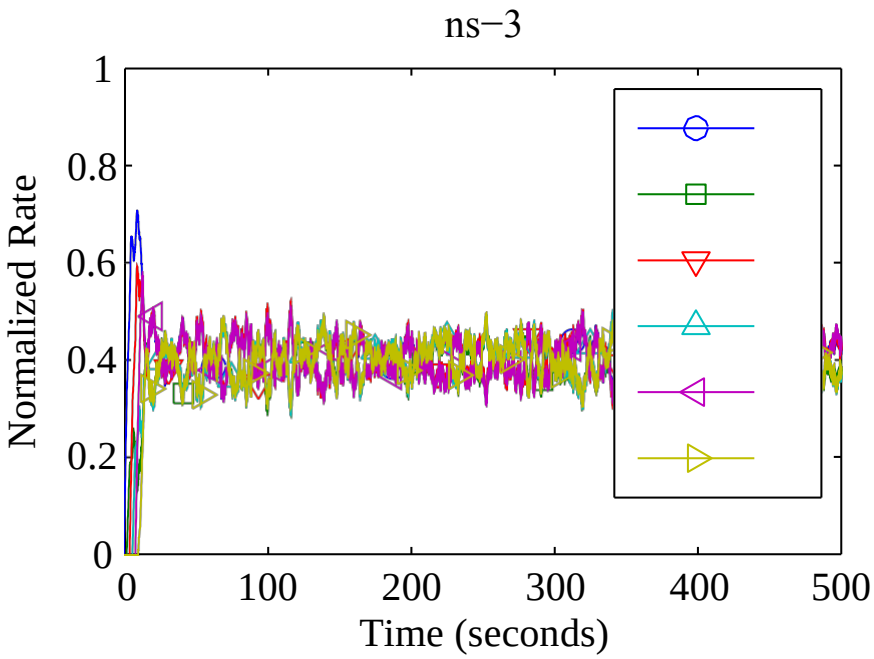


(b) ns-3

Figure 4: Model validation, 6-chain graph



(a) differential model



(b) ns-3

Figure 5: Model validation, 6-cycle graph

4.1.3 Wireless Network Mapping

Having modeled the IEEE 802.11 MAC using differential equations, we can now construct the signal structure of the network using a robust reconstruction process. The reconstruction process allows us to determine the DSF from noisy data collected from perturbation experiments on the wireless network. The procedure we utilize is a generalization of the one detailed in [18]. We first present a mathematical description of the general robust network reconstruction process before providing an algorithm that applies this process to a wireless network.

General Network Reconstruction Process

Solving the network reconstruction problem from noisy data means determining the correct structure of the Q matrix, which shows the interaction between measured links in the wireless network. If there are p links in the network, then Q has $p^2 - p$ unknowns and we want to quantify the smallest distance from the system's transfer function G to all possible boolean structures for Q, noting that there are 2^{p^2-p} of them. Small distances will correspond to candidates for the correct structure, while large distances will mean that the structures are highly unlikely to be the correct structure of the network.

There are a number of ways to model input-output data with noise and nonlinearities; we consider the additive uncertainty model. We examine the system $\Delta = G - X$, where X is the true system, G is the estimate of the system and Δ represents the unmodelled dynamics. Giving the system an input U yields $GU - XU$. Since $Y = GU$, the resulting system is $Y - XU$. Moreover, given that X is the true system, we know that $X = (I - Q)^{-1}P$, so the system then becomes $Y - (I - Q)^{-1}PU$. Multiplying on the left by $(I - Q)$ gives us $(I - Q)Y - PU$. Multiplying the system out gives us $Y - QY - PU$, where the values of Q and P are unknown.

The noisy version of this system can be solved using total least squares or the simpler least squares, which ignores the impact of noise on the operator, M, but is computationally simpler.

Algorithm for Robust Reconstruction of Wireless Networks

First, we note that a wireless network is reconstructible according to the necessary and sufficient informativity conditions provided in [1]. Since each link in a wireless network can be perturbed independently of the others, that the structure of P is diagonal, which is sufficient a priori information to make the system reconstructible, as noted in [5].

Second, before we must ensure that the network is in an equilibrium state, which can be achieved using rate controllers on the wireless network. The system must be in equilibrium

since we are applying linear tools to a nonlinear system, and we know the performance of a nonlinear system is similar to that of the linearized system close to equilibrium.

We now provide an algorithm for the reconstruction of wireless networks:

Step 1: Perturb each link ***i*** (usually with a step input), where each perturbation is large enough to escape the sensor or measurement noise, but small enough to ensure it remains in the same basin of attraction for the equilibrium chosen. After each perturbation, the rate on every link in the network should be measured until the transient response from the perturbation is complete and the system settles into a steady state.

Step 2: The third step is to solve the total least squares problem defined below for all 2^{p^2-p} boolean structures to determine the distance α_k for the k^{th} boolean structure. We could simplify this process further by restricting the search space of boolean structures to only include symmetric structures, since in wireless networks we consider interference to be biderectional, which would mean that the structure of Q is always symmetric.

The total least squares problem is defined as:

$$\min_{\vec{x}} || \left[\vec{\Delta_1} \quad \vec{e_1}\right] ||_\infty \tag{8}$$

subject to

$$\left[M + \Delta_1\right] \vec{x} = \vec{y} + e_1$$

Alternatively, one could solved a typical least squares problem:

$$\min_{\vec{x}} ||\vec{y} - M\vec{x}|| \tag{9}$$

which is simpler to compute, but sacrifices accuracy by ignoring the noise on the operator, M.

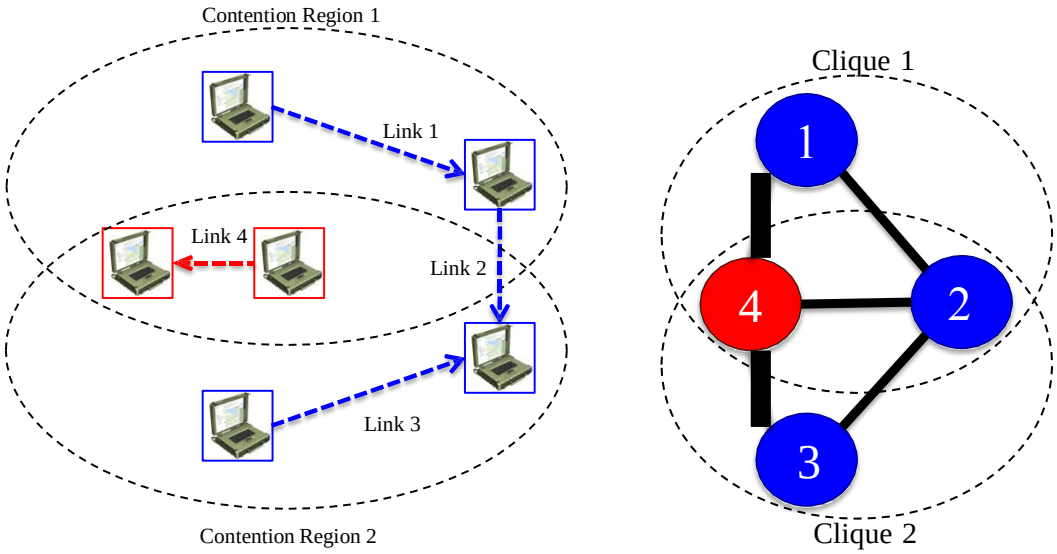
Step 3: Given an α_k corresponding to each boolean structure, use an information criterion, such as Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), or a variant (such as AICc), to select the DSF that best models the data. A full DSF can always fit the noise better and achieve a lower α score. To correct this problem, an information criterion weights the complexity of the DSF, in terms of number of edges in the DSF, to find a network structure that achieves a low α score with as few edges as possible. As a practical matter, one needs to tune the weights of the information criterion function to target the right degree of sparsity expected for any given application.

For simplicity, we assume a centralized controller can perform the perturbation experiments, collect the data, and run the reconstruction algorithm. Note that the algorithm works for any differential equation model of the 802.11 MAC, not just the one developed here.

For simplicity, we assume a centralized controller can perform the perturbation experiments, collect the data, and run the reconstruction algorithm. Note that the algorithm works for any differential equation model of the 802.11 MAC, not just the one developed here.

4.1.4 Transitory Interference Detection

We now use the differential model of the MAC presented in Section 7.1.2 to show that the DSF of the system changes when interfering devices are physically present in a wireless



(a) Topology (b) Contention graph

Figure 6: Example wireless network, with transitory devices shown in red, along with corresponding contention graph.

network. Whenever new wireless devices intrude on an existing network, the map changes because of the coupling it creates. Since the map is easy to create, whenever the performance of the links changes, we can recalculate the map in order to examine the network for possible interference from transitory devices. These devices could be other computers with WiFi interfaces or any device that emits an RF signal on the same frequency as the managed WiFi network.

Reconstruction Without Transitory Interference

Consider the wireless network as shown in Figure 6a, but without link 4 present. The contention graph for this network is given in Figure 6b, but without the node 4 and without any of the edges connected to node 4. We show how to derive the exact DSF for this network, assuming we know the differential equation model for the MAC presented in Section 7.1.2. This will provide a "ground truth" for the network, which the reconstruction algorithm should match. For this network, the equations describing the rates obtained by these links are given by:

$$\begin{bmatrix} \dot{b_1} \\ \dot{b_2} \\ \dot{b_3} \end{bmatrix} = \begin{bmatrix} u_1 - x_1 \\ u_2 - x_2 \\ u_3 - x_3 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} -x_1 + \beta_1 \sigma(b_1)\left(x_1 + (1-x_1)\left(1 - \frac{x_2}{1-x_1}\right)\right) \\ -x_2 + \beta_2 \sigma(b_2)\left(x_2 + (1-x_2)\left(1 - \frac{x_1}{1-x_2}\right)\left(1 - \frac{x_3}{1-x_2}\right)\right) \\ -x_3 + \beta_3 \sigma(b_3)\left(x_3 + (1-x_3)\left(1 - \frac{x_2}{1-x_3}\right)\right) \end{bmatrix}$$

Assume that a rate controller is being used on all the links to give fair network performance. The equilibrium rates for this network are $u_1 = 0.4$, $u_2 = 0.4$, and $u_3 = 0.4$.

(a) Before intrusion (b) After intrusion

Figure 7. Structure before and after intrusion.

We linearize the system of equations in (10) around this equilibrium to get:

$$\begin{bmatrix} \dot{x_1} \\ x_2 \\ x_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} -1 & -0.66 & 0 & 0.11 & 0 & 0 \\ -0.29 & -0.62 & -0.29 & 0 & 0.02 & 0 \\ 0 & -0.66 & -1 & 0 & 0 & 0.11 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

The exact (Q, P) for this system is:

$$Q = \begin{bmatrix} 0 & -\frac{66\,s}{100\,s^2+100\,s+11} & 0 \\ -\frac{29\,s}{2\left(50\,s^2+31\,s+1\right)} & 0 & -\frac{29\,s}{2\left(50\,s^2+31\,s+1\right)} \\ 0 & -\frac{66\,s}{100\,s^2+100\,s+11} & 0 \end{bmatrix}, \text{and}$$

$$P = \begin{bmatrix} \frac{11}{100\,s^2+100\,s+11} & 0 & 0 \\ 0 & \frac{1}{50\,s^2+31\,s+1} & 0 \\ 0 & 0 & \frac{11}{100\,s^2+100\,s+11} \end{bmatrix}.$$

A graphical representation of this DSF is shown in Figure 7a.

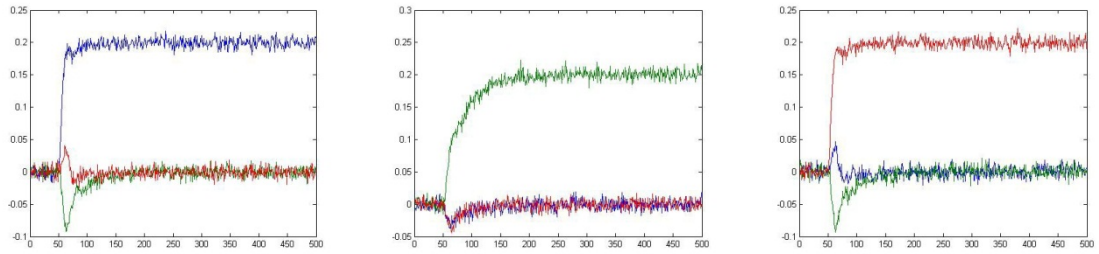
In practical situations, it is not possible to know the system of differential equations that describe the wireless network. In this case, we must use the robust reconstruction method described in Section 4.1.3 to determine the boolean structure of the DSF.

Step 1: Perturb each link and record the measurements of each perturbation experiment. The results of these experiments are shown in Figure 8.

Step 2: Using the noisy data that is collected, find the least squares solution, α_k, for all possible boolean structures. The results are in Table 1, along with their associated *AIC* values.

Step 3: Calculate the *AI C* values for each boolean structure using the equation:

$$AI\,C_k = 2L_k + 2In((2\pi\alpha_k) + 1) \tag{11}$$



(a) Perturbation on Link 1 (b) Perturbation on Link 2 (c) Perturbation on Link 3

Figure 8: Perturbation Experiments for Wireless Network without Transitory Interference

Table 1: Reconstruction Without Transient Interference

Results of Reconstruction Process		
Boolean Structure	**alpha**	**AIC**
0 1 1 1 0 1 1 0 0	1.22	14.32
0 1 0 1 0 1 0 1 0	0.14	9.262
0 1 1 1 0 1 0 1 0	0.09	10.9
0 1 0 1 0 1 1 1 0	0.08	10.81
0 1 1 1 0 1 1 1 0	0	12

where L_k are the number of nonzero elements in the k^{th} boolean structure.

In Table 1, we see that although the correct structure (marked in red) does not have the lowest α score, by using AIC we can discriminate against structures with more non-zero elements to determine the correct structure.

Reconstruction With Transitory Interference

Now let us assume that new network devices begin operating within the contention region of our original network, creating link 4 as shown in Figure 6a.

(b1,b2,b3,b4) = ((u1—x1) (u2-x2) (u3-x3) (u4-x4)) (12)

The new devices decide to set their load to $u_4 = 0.2$. This gives them the equilibrium sending rates of $x_4 = 0.2$. To keep the buffer from growing too large, link 2 has to recalibrate its sending rate to $u_2 = 0.2$ decreasing its rate to 0.2. Linearizing around the new equilibrium,

b_1	b_2	b_3	b_4	x_1	x_2	x_3	x_4
1	1.5	1	1.5	0.4	0.2	0.4	0.2

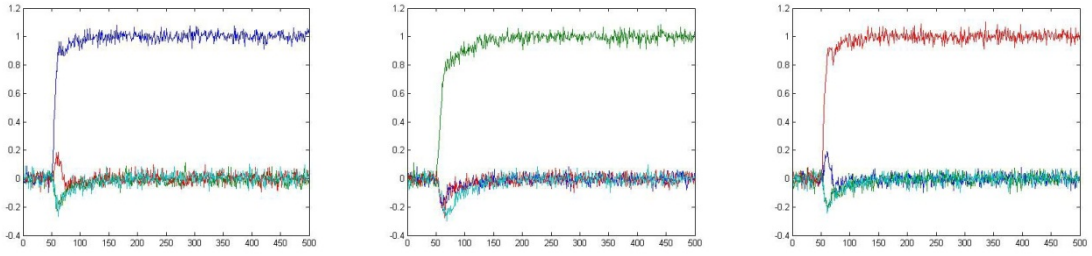
yields:

$$\begin{bmatrix} \dot{x_1} \\ x_2 \\ x_3 \\ x_4 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} -1 & -0.66 & 0 & -0.66 & 0.11 & 0 & 0 & 0 \\ -0.26 & -0.68 & -0.26 & -0.41 & 0 & 0.04 & 0 & 0 \\ 0 & -0.66 & -1 & -0.66 & 0 & 0 & 0.11 & 0 \\ -0.26 & -0.41 & -0.26 & -0.68 & 0 & 0 & 0 & 0.04 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

The new DSF, using the inputs on nodes 1, 2, and 3, is given by



(a) Perturbation on Link 1 (b) Perturbation on Link 2 (c) Perturbation on Link 3

Figure 9: Perturbation Experiments for Wireless Network with Transitory Interference

$$Q_2 = \begin{bmatrix} 0 & -\frac{33s\left(100s^2+27s+4\right)}{2(d(s))} & \frac{429s^2}{d(s)} \\ -\frac{26s}{100s^2+109s+4} & 0 & -\frac{26s}{100s^2+109s+4} \\ \frac{429s^2}{d(s)} & -\frac{33s\left(100s^2+27s+4\right)}{2(d(s))} & 0 \end{bmatrix},$$

and

$$P_2 = \begin{bmatrix} \frac{275s^2+187s+11}{d(s)} & 0 & 0 \\ 0 & \frac{2}{100s^2+27s+4}+\frac{2}{100s^2+109s+4} & 0 \\ 0 & 0 & \frac{275s^2+187s+11}{d(s)} \end{bmatrix}$$

where $d(s) = 2500s^4 + 4200s^3 + 1646s^2 + 287s + 11$.

Again, these dynamical structure functions (Q, P) have been determined directly from the differential equations. We will now assume that the differential equations that describe the wireless network are unknown, and we will use the robust reconstruction process from Section 4.1.3 to determine the boolean structure of the network while the interference is occurring.

Step 1: Perturb each link and record the measurements of each perturbation experiment. The results of these experiments are shown in Figure 9.

Step 2: Using the noisy data that is collected, find the least squares solution, α_k, for all possible boolean structures. The results are in Table 2, along with their associated *AIC* values.

Step 3: Calculate the *AIC* values for each boolean structure using Equation (11).

As we can now see, the correct structure, marked in red Table 2, is now the fully connected network. A graphical representation of this DSF is shown in Figure 7b. We can see that the links 1 and 3 were not coupled in Figure 7a, but now they appear coupled. This signals the presence of interfering wireless devices.

If the new devices do not create new links in the DSF, but do change the transfer functions contained in Q and P, then the presence of such an intruder is harder to detect.

4.2 Necessary and sufficient conditions for network reconstruction

Identifiability conditions fundamentally concern the definition of a map from model parameters to data and ensuring that it is injective. In this way, a particular set of parameters is

Table 2: Reconstruction With Transient Interference

Results of Reconstruction Process		
Boolean Structure	**alpha**	**AIC**
0 1 1 1 0 1 1 0 0	34.82	20.79
0 1 0 1 0 1 0 1 0	2.77	13.83
0 1 1 1 0 1 0 1 0	1.96	15.18
0 1 0 1 0 1 1 1 0	0.81	13.61
0 1 1 1 0 1 1 1 0	0.03	12.35

uniquely specified by the data, identifying the correct model from the set of models under consideration.

Identifying a system's DSF from data involves the standard issues related to identifying a TF from data (sufficiency of excitation, etc.), along with additional issues related to the fact that many DSF generate the same TF (consider Lemma 1). In the sequel we will ignore the standard issues and focus on the additional identifiability issues unique to DSF. Consequently we will assume that the system's TF has been successfully identified from data and focus on necessary and sufficient conditions for then recovering the DSF. To accomplish this, we will construct the map from the elements of the DSF to the associated TF and establish conditions ensuring this map is injective.

To facilitate the discussion, we introduce the following notation. Let $A \in C^{n\times m}$ and $B \in C^{k\times l}$. Then:

- blckdiag(A, B) is the block diagonal matrix given by

$$\begin{matrix} A & 0 \\ 0 & B \end{matrix} \quad ,$$

- a_i is the i^{th} column of matrix A,
- A_{-i} is the matrix A without it's i^{th}
- a_{ij} is the $(i, j)^{th}$ entry of matrix A,
- A' is the conjugate transpose of matrix A,
- $R(A)$ is the range of A,
- $\vec{a}$ is the vector stack of the columns of A, given by a1…am

- and $\overleftarrow{a}$ is the vector stack of the columns of A'.

We begin by rearranging the fundamental DSF relationship, $G = (I - Q)^{-1}P$ to yield:

$$\begin{bmatrix} I & G' \end{bmatrix} \begin{bmatrix} P' \\ Q' \end{bmatrix} = G' \tag{13}$$

Noting that

$AX = B \iff \text{blckdiag}(A, ..., A)x = b$

and defining $X = \begin{bmatrix} P^t & Q^t \end{bmatrix}$ then allows us to rewrite Equation (13) as

$$I \quad \text{blckdiag}(G', ..., G') \ \overleftarrow{x} = \overleftarrow{g}. \tag{14}$$

Further noting that since the diagonal elements of Q are identically zero and the dimensions of P, Q, and G are $p \times m$, $p \times p$, and $p \times m$ respectively, then exactly p elements of $\overleftarrow{x}$ are always zero. Abusing notation, we can then redefine $\overleftarrow{x}$ to remove these zero elements, reducing Equation (14) to the following:

$$I \quad \text{blckdiag}(G^t_{-1}, G^t_{-2}, ..., G^t_{-p}) \ \overleftarrow{x} = \overleftarrow{g}. \tag{15}$$

Equation (15) reveals the mapping from elements of the DSF, contained in $\overleftarrow{x}$, to its associated TF, represented by $\overleftarrow{g}$. The mapping is clearly a linear transformation represented by the matrix operator $I \quad \text{blckdiag}(G^t_{-1}, G^t_{-2}, ..., G^t_{-p})$. This matrix has dimensions $(pm) \times (pm + p^2 - p)$, and thus the transformation is certainly not injective. This is why not even the Boolean structure of a system's DSF can be identified – even from perfect information about the system's TF – without additional a priori structural information.

Identifiability conditions will thus be established by determining which elements of $\overleftarrow{x}$ must be known a priori in order to reduce the corresponding transformation to an injective map. To accomplish this, consider the $(pm + p^2 - p) \times k$ transformation T such that

$$\overleftarrow{x} = Tz \tag{16}$$

where z is an arbitrary vector of size k. The following lemma describes technical conditions on T establishing necessary and sufficient identifiability conditions for DSF reconstruction.

Lemma 1. *Let*

$$M = LT, \tag{17}$$

where $L = \begin{bmatrix} I & \text{blckdiag}(G'_{-1}, G'_{-2}, ..., G'_{-p}) \end{bmatrix}$ *and* T *is a* $(pm + p^2 - p) \times k$ *matrix operator as in Equation (16). Then* M *is injective if and only if*

1. $k \leq pm$, *and*

2. rank(T) $= k$ *(i.e. T is injective).*

Proof. Since $\boldsymbol{I}$ blckdiag($G^t_{-1}, G^t_{-2}, ..., G^t_{-p}$) has rank pm, rank(M) $=$ min(pm, rank(T)). If rank(T) $> pm$, implying $k > pm$, then M is clearly not injective. If rank(T) $\leq pm$, then rank(M) $=$ rank(T) and M will be injective if and only if $k =$ rank(T). □

Theorem 1. (Identifiability Conditions) *Given a system characterized by the transfer function G, its DSF (Q, P) can be identified if and only if*

1. *M, defined as in Equation (17), is injective, and*

2. $\overleftarrow{g} \in R(M)$.

Proof. The proof follows immediately from the observation that M is the mapping from unidentified model parameters to the system TF. Under these conditions one can clearly solve for z given G and then construct the DSF from $\overleftarrow{x}$, where $\overleftarrow{x} = Tz$, and T is precisely the a priori system information that is necessary and sufficient for reconstruction. □

We will now illustrate this reconstruction result on some simple examples.

Example 2. *Consider a system with square TF given by*

$$G = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1p} \\ G_{21} & G_{22} & & G_{2p} \\ \vdots & & \ddots & \vdots \\ G_{p1} & G_{p2} & \dots & G_{pp} \end{bmatrix}.$$

Previous work has shown that if G is full rank and it is known, a priori, that the control structure P is diagonal that reconstruction is possible. Here we validate that claim by demonstrating that the associated T matrix becomes:

$$\begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{21} \\ P_{22} \\ \vdots \\ P_{pp} \\ Q_{12} \\ \vdots \\ Qp(p-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

yielding the operator $M = LT$ as:

$$M = \begin{bmatrix} e_1 & 0 & 0 & G'_{-1} & \dots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & \dots & e_p & 0 & \dots & G' \end{bmatrix}$$

where e_i is a zero vector with 1 in the i^{th} position. Note that M is a square matrix with dimensions $p^2 \times p^2$ and will be invertible provided G is full rank, thus enabling reconstruction.

Example 3. *Given the following TF of a system:*

$$G = \begin{bmatrix} \frac{s+2}{s^2+3s+1} & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

We attempt to find the DSF (Q, P) of the system:

$$Q = \begin{matrix} 0 & Q_{12} \\ Q_{21} & 0 \end{matrix} \quad \text{and}$$

$$P = \begin{matrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{matrix}$$

yielding the vector of unknowns $x = P_{11} \quad P_{12} \quad P_{21} \quad P_{22} \quad Q_{12} \quad Q_{21}{}'$. This gives us the system of equations of the form $Lx = b$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{s+2}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 1 & 0 & 0 & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{s+2}{s^2+3s+1} \\ 0 & 0 & 0 & 1 & 0 & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{21} \\ P_{22} \\ Q_{12} \\ Q_{21} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{s+2}{s^2+3s+1} \\ -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} \\ \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

Without additional information a priori structural information, we can not reconstruct. Suppose, however, that we know a priori that P takes the form:

$$P = \begin{matrix} P_{11} & -P_{11} \\ 0 & P_{22} \end{matrix}$$

Note that this non-diagonal P fails to meet the previous conditions for reconstruction [6]. Nevertheless, the vector of unknowns x can then be decomposed into the form Tz as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \ \vec{z} = \begin{bmatrix} P_{11} & P_{22} & Q_{12} & Q_{21} \end{bmatrix}'$$

Replacing $\vec{x}$ with $T\vec{z}$ above yields the system of equations of the form $M\vec{z} = \vec{b}$, where $M = LT$:

$$\begin{bmatrix} 1 & 0 & \frac{s+2}{(s+1)(s^2+3s+1)} & 0 \\ -1 & 0 & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 0 & 0 & \frac{s+2}{s^2+3s+1} \\ 0 & 1 & 0 & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{22} \\ Q_{12} \\ Q_{21} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{s+2}{s^2+3s+1} \\ -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} \\ \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

In this case M is full rank, from theorem 1 we know that the system is reconstructible. By solving for $x = (M)^{-1}b$ we get the DSF to be:

$$Q = \begin{bmatrix} 0 & \frac{1}{s+2} \\ \frac{1}{s+1} & 0 \end{bmatrix} \quad and \quad P = \begin{bmatrix} \frac{1}{s+1} & -\frac{1}{s+1} \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

4.3 Vulnerability of open- and closed-loop systems

4.3.1 Attack Models

In the literature, attacks on control systems have been classified into two types: *denial of service attacks*, when the attacker jams a channel in order to destabilize the system, and *deception attacks*, when the attack adds perturbations on particular links in order to compromises the reliability of the controller's state estimates [11]. We consider a hybrid attack model where the attacker adds perturbations to the channels, not just jam them, in order to destabilize the system. We call this type of attack a *destabilizing attack*.

Denial of Service (DoS) Attack Denial of service attacks prevent signals from reaching their intended destination. This is probably the easiest and most common attack, and it is modeled as removal of an edge in an interconnected structure. It might be done by jamming the communication channel, disrupting the transmitter/receiver, changing the routing protocol, saturating the receiver with extraneous signals, etc. The attacker's intent of such an attack could be to degrade the system performance or to completely destabilize it. [10] shows that performance of networked control systems could decrease significantly under a DoS attack. [11] gives a method to find an optimal controller that minimizes the effect of such an attack on linear control systems.

In [12], the authors study whether a DoS attack on certain links can make the system unreachable or uncontrollable. They also develop graph theoretic algorithms to identify the minimal number of edges which are necessary for preserving controllability and observability.

Deception Attack The goal of a deception attack is to change the state estimates computed by a model-based controller. This type of attack is modeled as a stable additive perturbation to an edge in the network. All stabilizing controllers make the closed loop system stable, hence, a stabilizing controller is necessarily stabilizable from the plant. So, if at attacker gains access to the communication channel between the plant and the controller, state estimates of a model-based controller can be altered. To prevent this, many real systems such as power systems, sensor networks, etc., are equipped with a Bad Data Detector (BDD) [13, 8, 14]. A BDD, using the model of the plant, detects deviation of the state estimates from the expected and raises an alarm to notify the human operator. Because of the presence of measurement noise, this deviation is never zero, so the BDD ignores deviations that are smaller than a specified threshold. Hence, in the presence of BDDs, the attack has to change the state estimates without increasing the chance of raising an alarm.

In [13] the authors study this kind of attack in the context of a power system. They show that it is in fact possible for an attacker to change the state estimates to a specific value without increasing the chance of being detected. [14] studies a similar problem in the scenario of a wireless sensor networks. It maps an approximation of the set of all possible values the attacker could drive the estimates to.

[8] studies a slightly different problem. Here, the goal of the attacker is to change the estimate of one of the states without increasing the chance of being detected. The authors recognize that while doing this the attacker might want to use the fewest channels possible or might try to keep the magnitude of the attack signal small. For each type of attack, the authors then give a formulation of a *security index* of the system.

Destabilizing Attack Like deception attacks, these attacks effectively arise as an additive perturbation on a link in the system interconnection structure. Unlike deception attacks, however, they seek to destabilize the system rather than simply move the system state to a desired value without being detected. BDDs are clearly capable of detecting the destabilization resulting from such attacks, nevertheless serious damage and even complete plant shut-down may result by the time system operators are able to do anything about it.

A rich literature in systems and control theory explores the destabilization of systems due to additive perturbations, see for example [4] and the references therein. Security analysis of destabilizing attacks thus appears to be a robustness problem with respect to certain classes of perturbations. Indeed, we adopt this point of view, and consider security problems to be essentially robustness problems of various types.

The contribution of this work, then, applied to this class of attacks, is in the solution of a certain class of robustness problems over a particular kind of link model–corresponding to logical, rather than the physical, links of a system–and with respect to a specific class of perturbations. Unlike standard robustness measures that generally consider destabilizing perturbations acting over all channels and nodes of a system, here we restrict our attention specifically to perturbations that disrupt a single link in the system's signal structure. Our analysis then considers such single-link perturbations over all possible system links. In the next section we explore our link model in detail.

4.3.2 Link Models

The destabilizing attacks considered here are additive perturbations acting on a single link in a system's logical interconnection structure. There are many characterizations of a system's structure, see for example [15, 16]. One characterization would consider the interconnection structure among subsystems. This definition of structure, also called the system's subsystem structure, would represent the physical interconnection between physical components of a particular networked system. Under this notion of structure, a *link* would represent the signal passing between two subsystem nodes within the subsystem interconnection architecture. In contrast to the subsystem structure, this work considers another definition of system structure and, consequently, a different notion of a system link.

In this work, we consider a partition on signals of the system into two categories: exposed signals and hidden signals. The logical interconnection structure, or architecture–also called the system's signal structure–is the causal relationship between exposed signals in the system. In this definition of structure, a *link* is a system describing the causal dependency between

two exposed signal nodes of the logical interconnection architecture.

Some important consequences of this definition of link include the fact that a link may represent a very indirect and complicated pathway–through various hidden signals that may be components of other links in the system. Thus a link is associated with a particular set of dynamics–a system–that characterizes how the input signal is transformed into the output signal. The fact that hidden signals may be shared between links, however, is an important distinction between signal and subsystem interconnection structures. Note that a state of one subsystem, interconnected with others in a subsystem architecture (such as a standard feedback interconnection between two blocks), is never shared with other subsystems; the subsystem architecture effectively partitions the states of the networked system. In contrast, states on the links of the signal structure may, in fact, be shared with those of other links. This degree of abstraction is important for security problems because an additive perturbation on a link of the signal structure does not represent the corruption of a particular channel, as it would in the subsystem structure, but rather the idea that an attacker infiltrated a particular dependency between specific manifest variables.

The next section provides some background on Dynamical Structure Functions (DSF), which are used to represent the signal structure of a system. The DSF is a system representation that describes more structure, in the logical interconnection sense, than the transfer function provides, but less than the state space realization would reveal. Specifically, the DSF describes exactly the causal dependencies between manifest variables without offering any indication of the structure relating hidden variables. As a result, although every state space realization specifies a unique DSF, and every DSF specifies a unique transfer function, there are many DSF architectures consistent with any specific transfer function, and many state space realizations consistent with any specific DSF.

4.3.3 Background: Dynamical Structure Function

Before developing the main theorem, we will present a concise derivation of the dynamical structure function, and explain its relevance to the security of a networked system. For a complete derivation and results on different representations of structure see [16, 9, 6].

Let us consider a state-space LTI system

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} = \begin{bmatrix}\bar{A}_{11} & \bar{A}_{12}\\ \bar{A}_{21} & \bar{A}_{22}\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}\bar{B}_1\\ \bar{B}_2\end{bmatrix} u$$
$$y = \begin{bmatrix}\bar{C}_1 & \bar{C}_2\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix}, \qquad (18)$$

where $\bar{C}_1$ $\bar{C}_2$ has full row rank. This system can be transformed to:

$$\begin{bmatrix}\dot{y}\\ \dot{x}\end{bmatrix} = \begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}\begin{bmatrix}y\\ x\end{bmatrix} + \begin{bmatrix}B_1\\ B_2\end{bmatrix} u$$
$$y = \begin{bmatrix}I & 0\end{bmatrix}\begin{bmatrix}y\\ x\end{bmatrix}, \qquad (19)$$

Here y are the states that are measured, and x are the hidden states. Now, taking Laplace

Transforms of the signals in (19), we get

$$\begin{bmatrix} sY \\ sX \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U. \tag{20}$$

Solving for X in the second equation of (20) gives

$$X = (sI - A_{22})^{-1}A_{21}Y + (sI - A_{22})^{-1}B_2U$$

Substituting into the first equation of (20) we get,

$$sY = WY + VU,$$

where $W = A_{11} + A_{12}(sI - A_{22})^{-1}A_{21}$ and $V = A_{12}(sI - A_{22})^{-1}B_2 + B_1$. Let D be a diagonal matrix with the diagonal entries of W. Then,

$$(sI - D)Y = (W - D)Y + VU.$$

Now we can rewrite this equation as,

$$Y = QY + PU, \tag{21}$$

where

$$Q = (sI - D)^{-1}(W - D)$$

and

$$P = (sI - D)^{-1}V.$$

The matrix Q is a matrix of transfer functions from Y_i to Y_j, $i \neq j$, or relating each measured signal to the other measured signals. Note that Q is zero on the diagonal and either zero or a strictly proper transfer function on the off diagonal. The matrix P is a matrix of zeros or strictly proper transfer functions from each input to each output without depending on any additional measured states. Together, the pair $(Q(s), P(s))$ is called the *dynamical structure function* for system (18).

The transfer function matrix for this system is given by

$$G = (I - Q)^{-1}P = C(sI - A)^{-1}B.$$

Hence, G_{ij} is the closed loop transfer function from input j to state i. In this paper, we will also refer to the closed loop transfer function between states. A transfer function from state j to state i is represented by H_{ij}, where

$$H = (I - Q)^{-1}.$$

Note that the transfer function from a state to an input is always zero.

Definition 1. *Given a system 18 with signal structure characterized by the dynamical structure function (P,Q), a* link *(i, j) of the system corresponds to any nonzero entry in P or Q.*

Note that P gives the links from the inputs to the measured states, and Q gives the links that represent the dependencies between the measured states. The next section will introduce the notion of vulnerability and characterize vulnerable links in the system's architecture characterized by (P, Q).

4.3.4 Vulnerable Links

In this work, vulnerability refers to the destabilization of a system resulting from the corruption of a single link in its signal architecture. We begin with a definition of a vulnerable link.

Definition 2. *Given a system 18 with signal structure characterized by the dynamical structure function (P,Q), a link in (P,Q) is called* vulnerable *if there exists a stable perturbation on the link that makes the system unstable.*

Example 4. *Let us consider a system with*

$$P = \begin{bmatrix} \frac{1}{s+2} & 0 \\ 0 & \frac{1}{s+2} \end{bmatrix}, \text{ and } Q = \begin{bmatrix} 0 & \frac{1}{s+2} \\ \frac{1}{s+2} & 0 \end{bmatrix}.$$

This system is stable because the transfer function,

$$G = \frac{1}{s^2+4s+3}\begin{bmatrix} s+2 & 1 \\ 1 & s+2 \end{bmatrix}$$

Now let us add a perturbation $\Delta = \frac{3}{s+2}$ *to the link* Q_{12} *as shown in Figure 10. The resulting transfer function is G*

which is unstable. Hence the link Q_{12} *is a vulnerable link. Similarly, it can be shown that* Q_{21} *is vulnerable, although neither* P_{11} *nor* P_{22} *are vulnerable.*

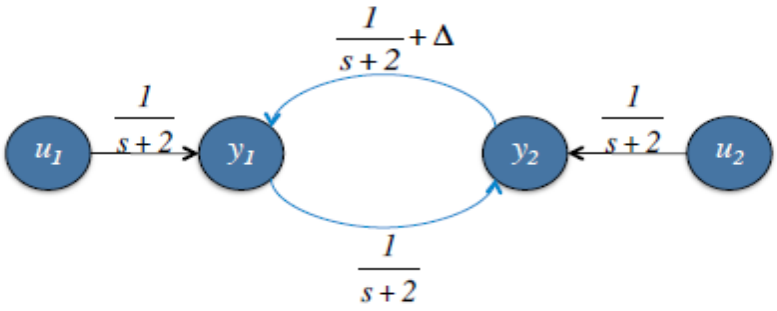


Figure 10: The system with the perturbation Δ. Black arrows indicate secure links, while blue arrows indicate vulnerable links.

Condition for Vulnerability Given that an attacker has the knowledge of the dynamical structure function representation of a system, we will derive a necessary and sufficient condition for a link to be vulnerable.

Theorem 2. *Let us consider a stable system (P, Q). There exists a stable additive perturbation* Δ *on a link from node* ***i*** *to node* ***j****, either in P or Q, that makes the system unstable if and only if the closed loop transfer function from node* ***j*** *to* ***i*** *is nonzero.*

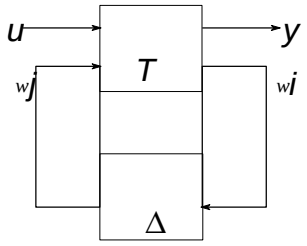


Figure 11: System with the perturbation $\Delta e_j e_i^T$

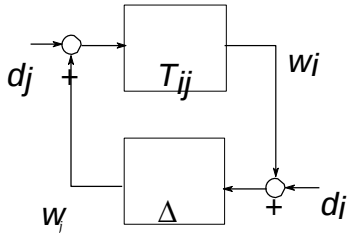


Figure 12: Necessary and sufficient condition for stability of the system in Figure 11

Proof. The system with the perturbation Δ can be represented as the linear fractional transformation in Figure 11, where T is the associated closed loop transfer function, and w_i, w_j represent the signals at node i and j respectively. This system is stable if and only if the system in Figure 12 is stable (see [4]). If $T_{ij} = 0$, any stable Δ does not affect the stability of the system in Figure 12. Thus the closed loop system in Figure 11 is stable for all Δ.

If $T_{ij} \not= 0$, then the system in Figure 12 is unstable if any of the transfer functions of $\begin{matrix} d_j \\ d_i \end{matrix} \rightarrow \begin{matrix} w_j \\ w_i \end{matrix}$ is unstable. We have,

$$w_j = \frac{1}{1 - T^{ij}\Delta} \begin{matrix} T_{ij} & \Delta \\ \Delta & \end{matrix} \begin{matrix} d_j \\ d_i \end{matrix}.$$

Let $T_{ij} = \frac{N}{D}$ and $\Delta = \frac{\delta_N}{\delta_D}$, then

$$w_j = \frac{D\delta_D}{D\delta_D - N\delta_N} \begin{matrix} \frac{N\delta_N}{D\delta_D} & \frac{\delta_N}{\delta_D} \\ \frac{\delta_N}{\delta_D} & \end{matrix} \begin{matrix} d_j \\ d_i \end{matrix}. \tag{22}$$

For a polynomial to be stable it is necessary that all its coefficients are of the same sign. In the case of the polynomial

$$R(s) = D\delta_D - N\delta_N, \tag{23}$$

it is easy to see that a properly designed Δ can zero out at least one of the terms. Thus, there exists a Δ that destabilizes these transfer functions. Note that when we are considering the vulnerability of the links in Q, $T = H = (I - Q)^{-1}$, gives the closed loop transfer functions. Now, we will present some implications of this result. □

Corollary 1. *None of the links in P are vulnerable.*

Proof. This is true because the transfer function from the states to the input is always zero. □

Corollary 2. *If T_{ij} is nonzero, there exists a perturbation $\Delta \in \mathrm{R}$ that destabilizes the system in Figure 12.*

Proof. Let $\Delta \in \mathrm{R}$, $\boldsymbol{l}_{ij} = \frac{N_I}{D_I}$. Thus, $\frac{\delta_n}{\delta_d} = \frac{\Delta D_I + N_I}{D_I}$, and the polynomial in (23) becomes $D_I D - N(D_I\Delta + N_I)$. We can see that at least one of the terms in this polynomial can be zeroed out by choosing appropriate Δ, making the polynomial unstable. □

Corollary 3. *Let us consider a stable system,*

$$\begin{aligned} \dot{x} &= Ax + \boldsymbol{I}u, \\ y &= \boldsymbol{I}x, \end{aligned} \tag{24}$$

where $A \in \mathrm{R}^{n\times n}$ and let $G = (sI - A)^{-1}$. There exists a perturbation $K = \Delta e_i e_j^T$, $\Delta \in \mathrm{R}$, such that $(A + K)$ is not Hurwitz, if and only if the transfer funtion from input u_i to output y_j, G_{ji}, is nonzero.

Proof. If the perturbation is on the diagonal entry of A, then it is easy to see that a destabilizing perturbation always exists and G_{ii} is never zero. Let $D = diag(A_{11}, A_{22}, ..., A_{nn})$.
The dynamical structure function of the system is given by $P = (sI - D)^{-1}$ and $Q = (sI - D)^{-1}(A - D)$. Any perturbation $K = \Delta e_i e_{\boldsymbol{j}}^T, \boldsymbol{i} \neq \boldsymbol{j}$ effects only the link Q_{ij}. Hence, the perturbation can make the system unstable if and only if the transfer function H_{ji} is nonzero. Also, the diagonal entries of P are nonzero, and $G = HP$. Thus, the transfer function H_{ij} is nonzero if and only if G_{ji} is nonzero.

□

Example 5. *Let us consider a system of the form (24) with*

$$A = \begin{bmatrix} -1 & 0 & -4 & 3 \\ 2 & -2 & 0 & 0 \\ 3 & 0 & -2 & -4 \\ 0 & 3 & -2 & -5 \end{bmatrix}$$

Here the eigenvalue of A are $\sigma = \{-1.5000 + 3.4278\boldsymbol{j}, -1.5000 - 3.4278\boldsymbol{j}, -6.7016, -0.2984\}$. Hence, the system is stable. In this system, the link from x_4 to x_1 is not vulnerable because $G_{41} = 0$. Notice that this example is not a trivial example, like a diagonal or a triangular system, since there are cycles that contain both nodes x_1 and x_4.

Corollary 4. *Let $A \in R^{n\times n}$. A perturbation on the $(i, \boldsymbol{j})^{th}$ entry of A changes its eigenvalues if and only if the $G_{ji} \neq 0$, where $G = (sI - A)^{-1}$ is the transfer function matrix i.e. the $(i, \boldsymbol{j})$ minor of $(sI - A)$ is nonzero.*

Proof. Take the system from Corollary 3. We can see that a perturbation on the $(i, \boldsymbol{j})^{th}$

entry has no effect on the system if $G_{ij} = 0$. Also, if $G_{ji} \neq 0$, the perturbation forms a closed loop system, such as the one given in Figure 12, in which case Δ definitely changes the poles of the system. □

If we take the A matrix from Example 5, note that its eigenvalues stay unchanged for any perturbation on the $(1, 4)^{th}$ entry.

Structure and Vulnerability To perform the vulnerability analysis of a system, we assume that the attacker can only modify existing links and cannot create new links. With this assumption, we can see that systems where the output nodes do not form a cycle are always secure, because in such a case the nodes can be permuted to obtain a triangular Q matrix. A triangular Q matrix gives a triangular H, and by applying Theorem 2 we can see that all the existing links are secure. Note that secure links doesn't always mean they are from a triangular system. For example, the link Q_{14} is secure in the system given in Figure 13, which is the signal structure architecture of the state-space system in Example 5.

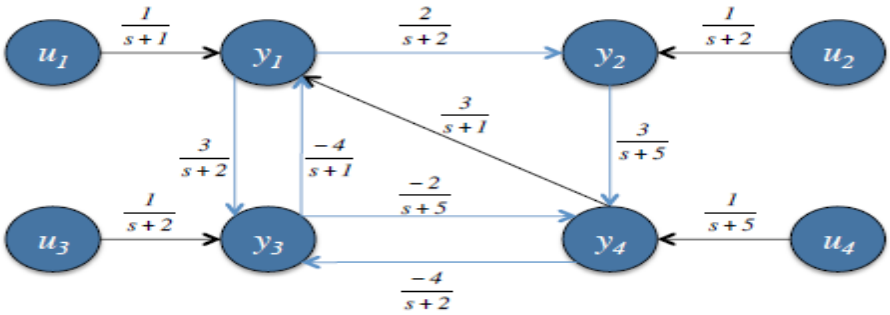


Figure 13: A system with a secure link in a cycle. Black arrows represent the secure links.

Noting that certain graphical structures result in secure links begs the question of whether there are particular dynamics that contribute to secure or vulnerable links in the system's architecture. The following theorem answers this question.

Theorem 3. *Every transfer function G has a completely secure architecture* $(\bar{P}, \bar{Q})$.

Proof. For any transfer function G, note that $(P = G, Q = 0)$ is an admissible Dynamical Structure Function since $G = (I - 0)^{-1}G$. From Corollary 1, we see that none of the links in P are vulnerable, and since Q has no links, the system is secure.

This result shows that the vulnerability of a system is structure dependent and not a function of the system dynamics. This fact highlights one difference between the vulnerability, which depends on the system structure and not the dynamics, and the robustness, which depends on the dynamics and not the system structure.

Measure of Vulnerability

Feedback is very common in both natural and engineered systems. Nevertheless, such structures usually generate vulnerable links. Thus, a measure of vulnerability is essential to understand the security of the system.

Given a signal architecture (P, Q) with associated closed loop transfer function T, the vulnerability of link (i, j) is given by

$$v_{ji} = ||T_{ij}||_\infty, \tag{25}$$

which is the inverse of the smallest perturbation required on link (i, j) to destabilize the system. Since all the links in P are secure, we only consider the links in Q while computing

the vulnerability, hence, $T = H$. The vulnerability of the system is given by

$$V = \max_{(i,j)\in Q} v_{ji} \tag{26}$$

$$= \max_{(i,j)\in Q} ||T_{ij}||_\infty \tag{27}$$

This measure allows us to associate a size of the smallest destabilizing perturbation with every link in the system architecture. Secure links thus have a vulnerability of 0. Note that V, the system vulnerability, is less than or equal to the inverse of the size of the smallest destabilizing perturbation for the system, since link perturbations are restricted to act on a single link only.

4.3.5 Numerical Example

Let us consider a system with the architecture given in Figure 14 where,

$$P = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 \\ 0 & \frac{1}{s+1} & 0 \\ 0 & 0 & \frac{1}{s+1} \end{bmatrix}$$

and

$$Q = \begin{bmatrix} 0 & 0 & \frac{1}{s+1} \\ \frac{1}{s+2} & 0 & 0 \\ 0 & \frac{1}{s+3} & 0 \end{bmatrix}$$

The transfer function matrix for the system is given by

$$G = \begin{bmatrix} \frac{s^3+6s^2+11s+6}{d(s)} & \frac{s+2}{d(s)} & \frac{s^2+5s+6}{d(s)} \\ \frac{s^2+4s+3}{d(s)} & \frac{s^3+6s^2+11s+6}{d(s)\, s^2} & \frac{s+3}{d(s)\, s^3+6s^2+11s+6} \\ \frac{s+1}{d(s)} & \frac{s^2+3s+2}{d(s)} & \frac{d(s)}{} \end{bmatrix},$$

where $d(s) = s^4 + 7s^3 + 17s^2 + 16s + 5$. By small gain theorem, the size of the smallest destabilizing perturbation is $||G||^{-1} = 0.42$ could destabilize the system.

Let $H = (I - Q)^{-1}$ represent the transfer function between the measured states y_i. Since the links in P are not vulnerable, we consider the perturbations on the links in Q which are the links (y_1, y_2), (y_2, y_3), and (y_3, y_1). To compute the vulnerability of these links we need the following transfer functions:

$$H_{12} = \frac{s+2}{s^3+6s^2+11s+5}$$

$$H_{23} = \frac{s+3}{s^3+6s^2+11s+5}$$

$$H_{31} = \frac{s+1}{s^3+6s^2+11s+5}.$$

For this system $v_{21} = 0.4$, $v_{32} = 0.6$, and $v_{13} = 0.2$. Hence, $V = v_{23} = 0.6 < ||G||_\infty$, and the smallest perturbation on a single link that can destabilize this system must have a gain of $\frac{1}{V} = 1.67$.

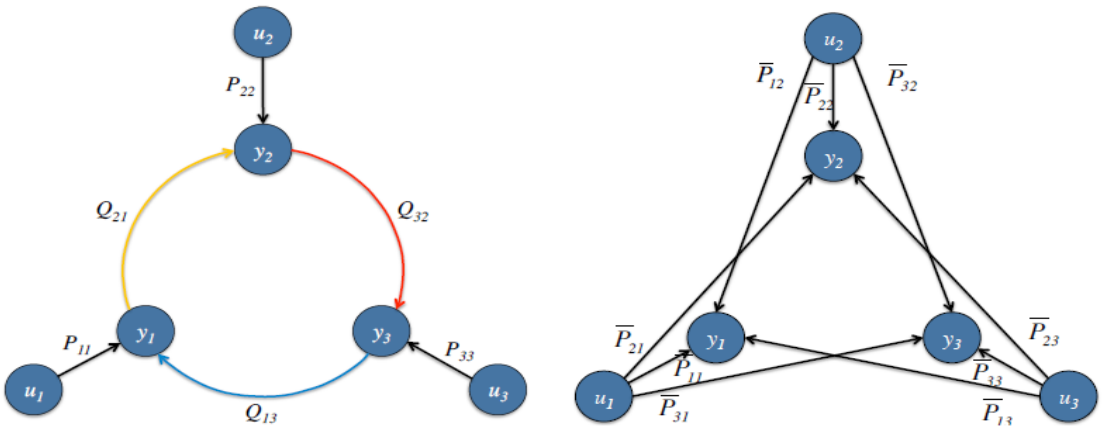


Figure 14: Vulnerable and secure architectures for the same transfer function. Black links are secure, vulnerable links are colored blue, yellow, and red in the increasing order of their vulnerability.

This system can also be implemented as shown in Figure 14b, where $\overline{P} = G$. This is one of the secure implementations of the system in Figure 14a. From this example we thus observe the following:

- The same transfer function can exhibit both vulnerable and secure architectures,
- System robustness, characterized by the size of the smallest destabilizing perturbation (0.42 in this example), is not equivalent to the inverse of the system vulnerability, characterized by the size of the smallest destabilizing perturbation on a single link (about 1.67 in this example),
- Only links in Q can be vulnerable.

4.4 Design of stabilizing distributed controllers with arbitrary signal structure constraints

In this section, we present a procedure to design a controller (Q, P) with a structure given by (Q^{bin}, P^{bin}) to stabilize a plant with the transfer function matrix G. It is an iterative procedure; we add the controller links to the plant such that the additional link attempts to stabilize the plant as well as all the previously added controller links. The procedure is as follows:

Procedure P

1. Choose an undesigned link p_{ij} such that $p_{ij}^{bin} = 1$
2. Design p_{ij} to stabilize g_{ji} such that there is no pole zero cancellation in PG. That is, the controller link is designed such that it stabilizes the transfer function it sees, and there is no pole-zero cancellation

3. After adding p_{ij}, if the closed loop system (G, P) is still unstable, repeat for all p_{xy}, $p_{xy}^{bin} = 1$, so that the added link attempts to stabilize the plant as well as all the previously added controller links.

4. If the closed loop system S, formed by adding P in feedback with G, is still unstable, add links in Q^{bin} such that there is no pole-zero cancellation between Q and S. Again, each added link attempts to stabilize the plant G along with the previously added links of P and Q.

Theorem 4. *Given a transfer function matrix, G, and a desired signal structure for a feedback controller characterized by* (Q^{bin}, P^{bin})*, Procedure* P *either delivers a stabilizing controller with the desired structure or no such controller exists.*

This theorem says that if the controller obtained using this procedure does not stabilize the plant, then there is no controller of the given structure that can stabilize it. Hence, this procedure provides a test for the existence of a structured stabilizing controller, and if such a controller exists, it synthesizes a nominal stabilizing controller that meets the structural constraint. Before proving this theorem, we will prove some lemmata.

Lemma 2. *Let K be the controller transfer function. A link* k_{ij} *cannot affect a mode of the plant G that is not observable or controllable from this link.*

Proof. Let,

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \text{ and } k_{ij} = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & 0 \end{array}\right].$$

Since we are only adding one link, both of these systems are SISO. Using the Kalman decomposition on G, we can transform it such that

$$A = \begin{bmatrix} A_{co} & 0 & A_{\times o} & 0 \\ A_{c\times} & A_{c\bar{o}} & A_{\times\times} & A_{\times\bar{o}} \\ 0 & 0 & A_{\bar{c}o} & 0 \\ 0 & 0 & A_{\bar{c}\times} & A_{\bar{c}\bar{o}} \end{bmatrix}, B = \begin{bmatrix} B_{co} \\ B_{\bar{c}o} \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} C_{co} & 0 & C_{c\bar{o}} & 0 \end{bmatrix}, \text{ and } D = d.$$

Here, the eigenvalues of $A_{c\bar{o}}$, $A_{\bar{c}o}$, and $A_{\bar{c}\bar{o}}$ are the modes of G that are unobservable, uncontrollable, and both respectively from feedback link k_{ij}.

The closed loop modes are given by the eigenvalues of the following matrix:

$$A_{cl} = \begin{bmatrix} A & BC_k \\ B_kC & A_k + B_kDC_k \end{bmatrix}$$

$$= \begin{bmatrix} A_{co} & 0 & A_{\times o} & 0 & B_{co}C_k \\ A_{c\times} & A_{c\bar{o}} & A_{\times\times} & A_{\times\bar{o}} & B_{c\bar{o}}C_k \\ 0 & 0 & A_{\bar{c}o} & 0 & 0 \\ 0 & 0 & A_{\bar{c}\times} & A_{\bar{c}\bar{o}} & 0 \\ B_kC_{co} & 0 & B_kC_{c\bar{o}} & 0 & A_k + B_kDC_k \end{bmatrix}$$

Transforming this matrix using the permutation

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

we get,

$$A_{clT} = TA_{cl}T' = \begin{bmatrix} A_{c\bar{o}} & A_{c\times} & B_{c\bar{o}}C_k & A_{\times\bar{o}} & A_{\times\times} \\ 0 & A_{co} & B_{co}C_k & 0 & A_{\times o} \\ 0 & B_kC_{co} & A_k + B_kDC_k & 0 & B_kC_{c\bar{o}} \\ 0 & 0 & 0 & A_{\bar{c}\bar{o}} & A_{\bar{c}\times} \\ 0 & 0 & 0 & 0 & A_{\bar{c}o} \end{bmatrix}$$

We can see that A_{clT} is block triangular, and the uncontrollable or unobservable modes, namely the eigenvalues of $A_{\bar{c}o}$, $A_{c\bar{o}}$, and $A_{\bar{c}\bar{o}}$, are not affected by the choices of A_k, B_k, or C_k. □

This result shows that when a controller link is added to the system such that it stabilizes all the modes that it can control and observe, it cannot destabilize other modes of the system that are already stable. Now, the following lemma gives a necessary and sufficient condition for the existence of the controller with transfer function structure K^{bin}.

Lemma 3. *There exists a controller with pattern K^{bin} that stabilizes a plant G if and only if every unstable mode of G is controllable and observable from at least one link k_{ij}, $k_{ij}^{bin} = 1$.*

Proof. From Lemma 2, we know that a link in the feedback controller cannot affect the uncontrollable or unobservable modes. Hence, any controller that stabilizes a given G must have links such that all the unstable modes are both controllable and observable from at least one of the controller link. Also, if every unstable mode is controllable and observable from some controller links, these links can stabilize the plant. □

lemmata 2 and 3 allow us to add links in P, since adding a link in P cannot change the controllability/observability of the plant for the other links in P. However, adding these links might cause the links in Q to lose controllability or observability of some of the modes, because links in Q are added on top of the links in P. Also, the links in Q themselves can create controllability/observability issues for subsequent links in Q.

Loss of observability/controllability can happen for two reasons: structurally or by exact cancellations. If it happens because of structural reasons, the system stays uncontrollable/unobservable for any choice of P or Q as long as it has the same structure. However, if the problem occurs because of exact cancellations, we can avoid these issues by a proper choice of the transfer function. Lemma 4 provides a methodology to design P and Q such that these cancellations are prevented. We will use the following result from [2] to prove the lemma.

Theorem 5. *Let G, H be proper rational transfer function matrices and suppose that $\det[\boldsymbol{I} + G(\infty)H(\infty)] \neq 0$. Then all the poles of the transfer function matrix*

$$W = \begin{matrix} (\boldsymbol{I} + HG)^{-1} & -H(\boldsymbol{I} + GH)^{-1} \\ G(\boldsymbol{I} + HG)^{-1} & (\boldsymbol{I} + GH)^{-1} \end{matrix}$$

are stable if and only if

- *GH has no unstable pole-zero cancellation, and*
- *all the poles of $(\boldsymbol{I} + GH)^{-1}$ are stable.*

Proof. See [2] Theorem 5. □

Lemma 4. *Loss of controllability/observability can be prevented from each link in Q if pole-zero cancellations are avoided in PG and QS. Here, S is the closed loop transfer function that Q observes and controls.*

Proof. The transfer function that Q observes for the closed loop system formed by adding P in feedback with G is given by $S = (\boldsymbol{I} - PG)^{-1}$. Using the Theorem 5, since there is no pole zero cancellations in PG, the closed loop system is stable if and only if S is stable. Which says that this transfer function has all the poles of the system. Hence Q observes and controls all the poles of the system after adding all the links in P if there is no pole zero cancellation in PG.

Similarly, when adding the links in Q if there is no pole zero cancellation in QS the controllability and observability properties are maintained. That is, if a mode is observable/controllable from a link Q_{ij} for some choices of the other links in the controller, then choosing the links in this fashion will keep the mode observable/controllable from Q_{ij}. □

Now we will present the proof of Theorem 1:

Proof. For every controller link that is added, either in P or Q, it stabilizes all the modes that are controllable and observable. Also, by Lemma 2, a newly added link cannot destabilize a mode that was already stable. Hence with every new link added to the system, the number of unstable modes either decreases or stays the same.

If every unstable mode in the system is controllable and observable by some link, it gets stabilized. If the plant has an unstable mode that is uncontrollable and unobservable from every link in P and Q, then by Lemma 3, there is no controller with the given pattern that stabilizes the plant. Also, since the added links satisfy the conditions in Lemma 4, if a mode is controllable or observable from a link for any choices previously added links, then it is controllable and observable. □

5 Conclusions

Our work demonstrates that Dynamical Structure Functions can be used to map the interference relationships of links in a wireless network. This method uses a model we have developed for the IEEE 802.11 MAC that uses differential equations to capture its dynamics

and equilibrium behavior. An advantage of this method is that it can be used to detect transitory interference from devices that are not a part of the managed network.

Two important aspects of our reconstruction algorithm are that the network must reach equilibrium for each perturbation test, and that the tests must be coordinated by a centralized controller. As part of our future work we will use experimental validation to explore how quickly equilibrium can be reached and the overhead required to conduct the experiments. We also want to test the impact of the tests on live traffic, to ensure they are non-invasive. We are also investigating how to distribute the reconstruction algorithm. Finally, we are interested in determining how we can use DSF to build an optimal rate controller for wireless networks.

We also considered a number of theoretical results, namely the necessary and sufficient informativity conditions for network reconstruction, a theory of vulnerability for open- and closed-loop systems, and a heuristic for designing decentralized controllers to satisfy an arbitrary signal structure constraint. Each of these results suggest new questions about the nature of network structure on a networks performance and robustness. Future work will explore these results in more detail, extending them to the nonlinear setting and introducing them to a variety of applications.

6 References

[1] J. Adebayo, T. Southwick, V. Chetty, and E. Yeung. Dynamical structure function identifiability conditions enabling signal structure reconstruction. In *IEEE Conference on Decision and Control*, 2012.

[2] B. Anderson and M. Gevers. On multivariable pole-zero cancellations and the stability of feedback systems. *IEEE Transactions on Circuits and Systems*, August 1981.

[3] Nicola Baldo, Manuel Requena-Esteso, José Núñez Martínez, Marc Portolès-Comeras, Jaume Nin-Guerrero, Paolo Dini, and Josep Mangues-Bafalluy. Validation of the IEEE 802.11 MAC model in the ns3 simulator using the EXTREME testbed. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010.

[4] G. E. Dullerud and F. Paganini. *A course in robust control theory, a convex approach*. Springer, 2000.

[5] J. Goncalves, R. Howes, and S. Warnick. Dynamical structure function for the reverse engineering of LTI networks. In *IEEE Conference on Decision and Control*, 2007.

[6] J. Goncalves and S. Warnick. Necessary and sufficient conditions for dynamical structure reconstruction of lti networks. *IEEE Transactions on Automatic Control*, August 2008.

[7] J. Goncalves and S. Warnick. *System-Theoretic Approaches to Network Reconstruction*, chapter 13, pages 265 – 295. Control Theory and Systems Biology, MIT Press, Eds. B. Ingalls and P. Iglesias, 2009.

[8] A. Teixeira H. Sandberg and K. H. Johansson. On security indices for state estimators in power networks. In *First Workshop on Secure Control Systems*, 2010.

[9] R. Howes. Application and properties of dynamical structure functions. Undergraduate honors thesis, Brigham Young University, 2008.

[10] J. Y. Hung M. Long, C.H. Wu. Denial of service attacks on network-based control systems: Impact and mitigation. *IEEE Transactions on Industrial Informatics*, 1(2), May 2005.

[11] A. Cardenas S. Amin and S. S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In *Hybrid Systems: Computation and Control*, 2009.

[12] D.D. Siljak V. Pichai, M.E. Sezer. Vulnerability of dynamic systems. *International Journal of Control*, 1981.

[13] P. Ning Y. Liu and M. Reiter. False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM conference on computer and communications security*, 2009.

[14] A. Casavola B. Sinopoli Y. Mo, E. Garone. False data injection attacks against state estimation in wireless sensor networks. In *Proceedings of the Conference on Decision and Control*, 2010.

[15] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick. Representing structure in linear interconnected dynamical systems. In *Proceedings of Conference on Decision and Control*, 2010.

[16] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick. Mathematical relationships between representations of structure in linear interconnected dynamical systems. In *Proceedings of American Control Conference*, 2011.

[17] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick. Representing structure in linear interconnected dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2011.

[18] Y. Yuan, G.B. Stan, S. Warnick, and J. Goncalves. Robust dynamical network reconstruction. *Automatica, special issue on Systems Biology*, 2011.

7 List of Acronyms

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
DOS	Denial of Service
DSI	Dynamical Structure Function
IP	Internet Protocol
MAC	Media Access Control
NS	Network Simulation
RF	Radio Frequency
TCP	Transmission Control Protocol
TF	Transitory Interference
UDP	User Data Protocol