On Analyzing Self-Driving Networks: *A **Systems Thinking** Approach*

Touseef Yaqoob, Muhammad Usama,
Junaid Qadir, Gareth Tyson

Associate Professor,
Information Technology
University (ITU), Pakistan

Session Outline

1. The Problems: Motivating Systems Thinking
2. The Solution: What Is Systems Thinking?
3. Systems Thinking For The Internet and Future AI-Driven SDNs

1

Motivating Systems Thinking

A system is always composed of
***elements**, **interconnections**, and **purpose**.*

Properties of Complex Adaptive Systems

Tightly Coupled

"Everything influences everything else";

"You can't just do one thing"

Dynamic

Change occurs at many time scales

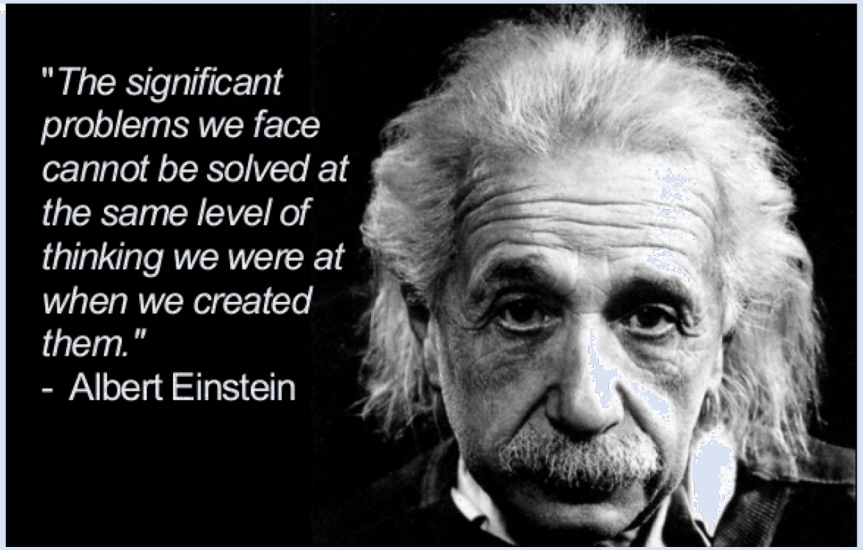
Exhibit Tradeoffs

Long term behavior is often different from short term behavior

Counterintuitive

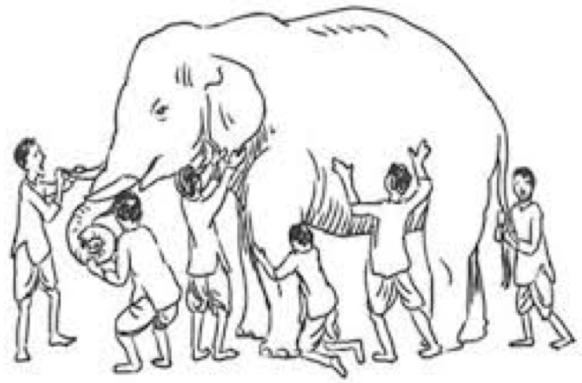
Cause and effect are distant in time and space

Problems Due to Non-Systemic Thinking in Complex Adaptive Systems

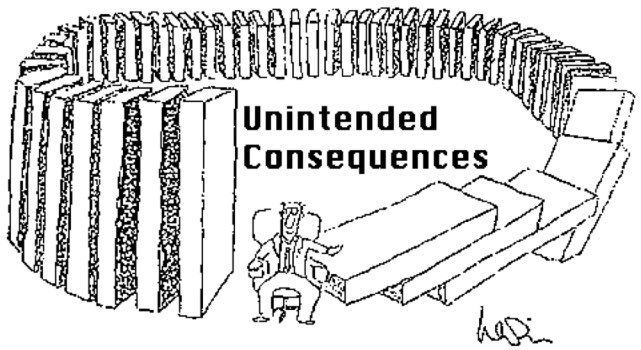


Mental models

dictates how we perceive reality based on our values, expectations, & experiences.

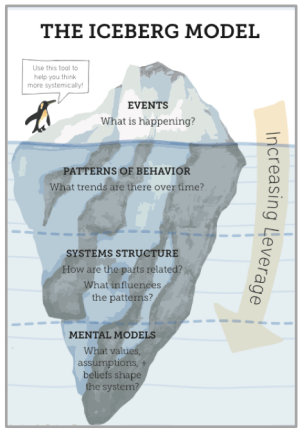


Unintended Consequences



Symptom Treatment

"There's always an easy solution to every problem that is neat, and plausible, but wrong."— H. L. Mencken



Optimizing the parts rather than the whole

Policy Resistance

measures taken to improve a situation can directly make it worse due to policy resistance arising from people adapting

2

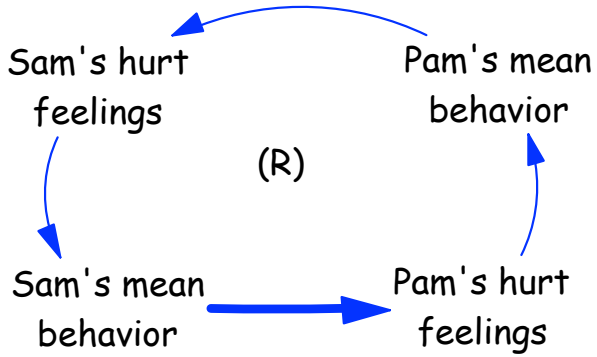
What Is Systems Thinking?

"Systems Thinking is the art and science of linking structure to performance, and performance to structure—often for purposes of changing structure (relationships) so as to improve performance"—Richmond.

Open loop vs. closed loop thinking
System-as-a-cause thinking

Every influence is both a cause and an effect (due to feedbacks)

"If you are not aware of how you are part of the problem, you can't be part of the solution."



Exogenous point of view

Sam is always mean to Pam.

It's all his fault. If he would be nicer, Pam's life would be better.

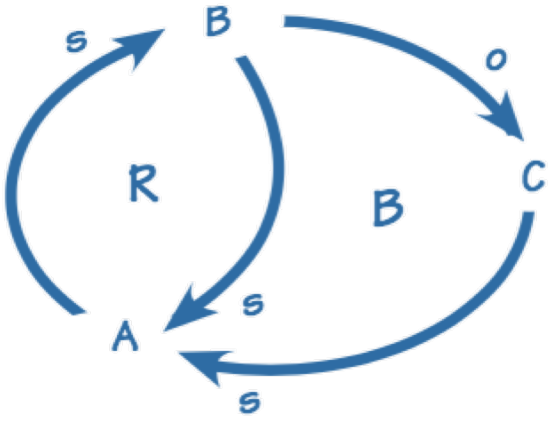
Endogenous point of view

Maybe there is something Pam is doing, which is causing Sam to be mean…

Exogenous: originating externally; ***Endogenous:*** originating internally

System thinking tools

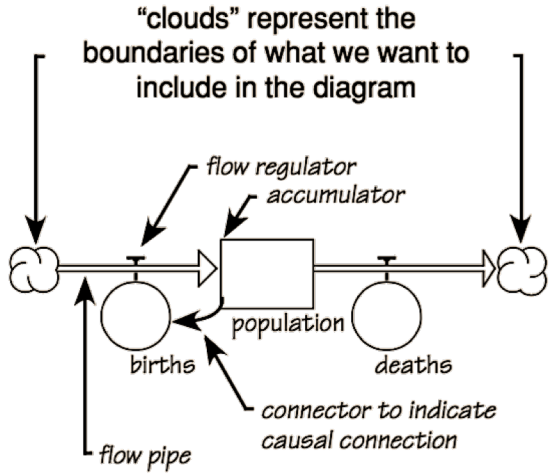
Qualitative tools



Causal loop diagrams

a framework for seeing interrelationships rather than things; can help in identifying reinforcing (R) and balancing (B) processes.

Quantitative tools



Stock and flow diagrams

Unlike causal loops, stock and flow diagrams provide information about rates of change and accumulations.

System dynamics is grounded in control theory and the modern theory of nonlinear dynamics and offers many other rigorous tools

3

Systems Thinking For The Internet and Future AI-Driven Self-Driving Networks

Why (and How) Networks Should Run Themselves

Nick Feamster and Jennifer Rexford
Princeton University

arXiv:1710.11583v1 [cs.NI] 31 Oct 2017

Abstract

The proliferation of networked devices, systems, and applications that we depend on every day makes managing networks more important than ever. The increasing security, availability, and performance demands of these applications suggest that these increasingly difficult network management problems be solved in real time, across a complex web of interacting protocols and systems. Alas, just as the importance of network management has increased, the network has grown so complex that it is seemingly unmanageable. In this new era, network management requires a fundamentally new approach. Instead of optimizations based on closed-form analysis of individual protocols, network operators need data-driven, machine-learning-based models of end-to-end and application performance based on high-level policy goals and a holistic view of the underlying components. Instead of anomaly detection algorithms that operate on offline analysis of network traces, operators need classification and detection algorithms that can make real-time, closed-loop decisions. Networks should learn to drive themselves. This paper explores this concept, discussing how we might attain this ambitious goal by more closely coupling measurement with real-time control and by relying on learning for inference and prediction about a networked application or system, as opposed to closed-form analysis of individual protocols.

1 Introduction

Modern networked applications operate at a scale and scope we have never seen before. Virtual and augmented reality require real-time responsiveness, micro-services deployed using containers introduce rapid changes in traffic workloads, and the Internet of Things (IoT) significantly increases the number of connected devices while also raising new security and privacy concerns. The widespread integration of these applications into our daily lives raises the bar for network management, as users elevate their expectations for real-time interaction, high availability, resilience to attack, ubiquitous access, and scale. Network management has always been a worthwhile endeavor, but now it is mission critical.

Yet, network management has remained a Sisyphean task. Network operators develop and use scripts and tools to help them plan, troubleshoot, and secure their networks, as user demands and network complexity continue to grow. Networking researchers strive to improve the tuning, design, and measurement of network protocols, yet we continue to fall behind the curve, as the protocols, variable network conditions, and relationships between them and user quality of experience become increasingly complex. Twenty years ago, we had some hope of (and success in) creating clean, closed-form models of individual protocols, applications, and systems [4, 24]; today, many of these are too complicated for closed-form analysis. Prediction problems such as determining how search query response time would vary in response to the placement of a cache are much more suited to statistical inference and machine learning based on measurement data [29].

Of course, we must change the network to make network management easier. We have been saying this for years, as we continue to fall behind the curve. Part of the problem, we believe, is the continued focus on designing, understanding, and tweaking individual protocols—we focus on better models for BGP, optimizations for TCP, QUIC, DNS, or the protocols du jour. In fact, our troubles do not lie in the protocols. The inability to model holistic network systems, as opposed to individual protocols, has made it difficult for operators to understand what is happening in the network. Software-Defined Networking (SDN) helps by offering greater programmability and centralized control, yet controller applications still rely on collecting their own data and installing low-level match-action rules in switches and SDN does not change the fact that real networked systems are too complex to analyze with closed-form models.

As networking researchers, we must change our approach to these problems. An ambitious goal for network management is that of a *self-driving network*—one where (1) network measurement is task-driven and tightly integrated with the control of the network; and (2) network control relies on learning and large-scale data analytics of the entire networked system, as opposed to closed-form models of individual protocols. Recent initiatives have proffered this high-level goal [14, 28], drawing an analogy to self-driving cars, which can make decisions that manage uncertainty and mitigate risk to achieve some task (e.g., transportation to some destination). This paper explores this goal in detail, developing the technical requirements for and properties of a self-driving network and outlining a broad, cross-disciplinary research agenda for the community that can move us closer to realizing this goal.

The networking research community has been developing the pieces of this puzzle for many years, from predictive models of application performance [18, 29] to statistical anomaly and intrusion detection algorithms based on analysis of network traffic [2, 7]. The state of the art, however, merely lays the foundation for the much more ambitious agenda of

1

new approach. Instead of optimizations based on closed-form analysis of individual protocols, network operators need data-driven, machine-learning-based models of end-to-end and application performance based on high-level policy goals and a holistic view of the underlying components. Instead of anomaly detection algorithms that operate on offline analysis of network traces, operators need classification and detection algorithms that can make real-time, closed-loop decisions. Networks should learn to drive themselves. This paper

① ②

Endogenous causes of Internet's problem

Paradoxes of Internet Architecture

Srinivasan Keshav
University of Waterloo

The top level goal for the DARPA Internet Architecture was to develop an effective technique for multiplexed utilization of existing interconnected networks.[3]

This goal is then further elaborated to encompass the following seven subgoals:

1. *Internet communication must continue despite loss of networks or gateways.*
2. *The Internet must support multiple types of communications service.*
3. *The Internet architecture must accommodate a variety of networks.*
4. *The Internet architecture must permit distributed management of its resources.*
5. *The Internet architecture must be cost effective.*
6. *The Internet architecture must permit host attachment with a low level of effort.*
7. *The resources used in the Internet architecture must be accountable.*

It is important to understand that these goals are in order of importance, and an entirely different network architecture would result if the order were changed.[3]

"Systems are perfectly designed to achieve the results they are currently achieving."—Deming

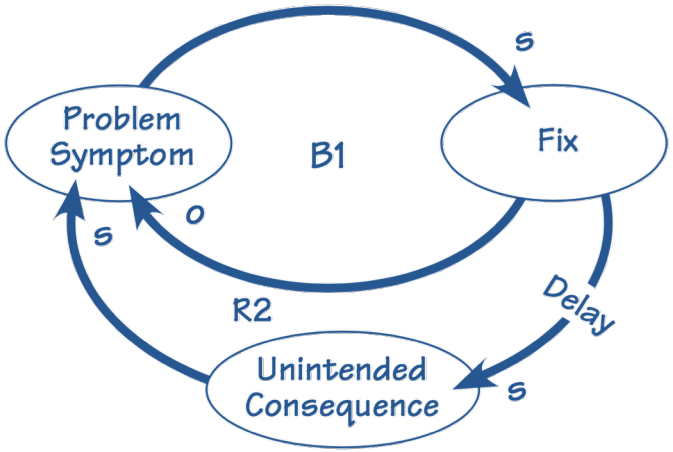
Three fundamental problems with the Internet today.

1. Spam.
2. Privacy and Security
3. Quality of Service

Keshav points out that these problems stem out from the same Internet's architectural elements responsible for its success.

System archetypes

System archetypes *are feedback structural templates that can be used for diagnosing vexing long-term problems.*



Fixes That Backfire
A quick solution with unexpected long-term consequences

IPv4 NAT; Bufferbloat

Limits to Growth
Improvement accelerates and then suddenly stalls

IPv4

Shifting the Burden
Systems unconsciously favor short-term, addictive solutions

IPv4 NAT; cross-layer design;
Tussles in cyberspace

Tragedy of the Commons
Shared unmanaged resource collapses due to overconsumption

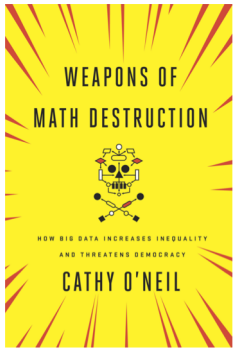
Spectrum commons;

Success to the Successful
Things get better for "winners" and worse for "losers"

Network neutrality
Walled gardens

Ethical and security policy challenges

“ THERE ARE ETHICAL CHOICES IN EVERY SINGLE ALGORITHM WE BUILD



The question of agency—i.e., **“who will take the ethical decision?”**— also looms large for self-driving networks.

“No problem stays solved in a dynamic environment.”—*Russell Ackoff* ”

Concluding remarks

Systems thinking helps us make sense of interdependency in complex system and the holistic behavior of a system by understanding the feedback loops at play.

With the rise of interest in self-driving networks, which will become part of the larger Internet, there is a need to rigorously look at how these technologies will affect—positively as well as negatively—all the stakeholders.

junaid.qadir@itu.edu.pk