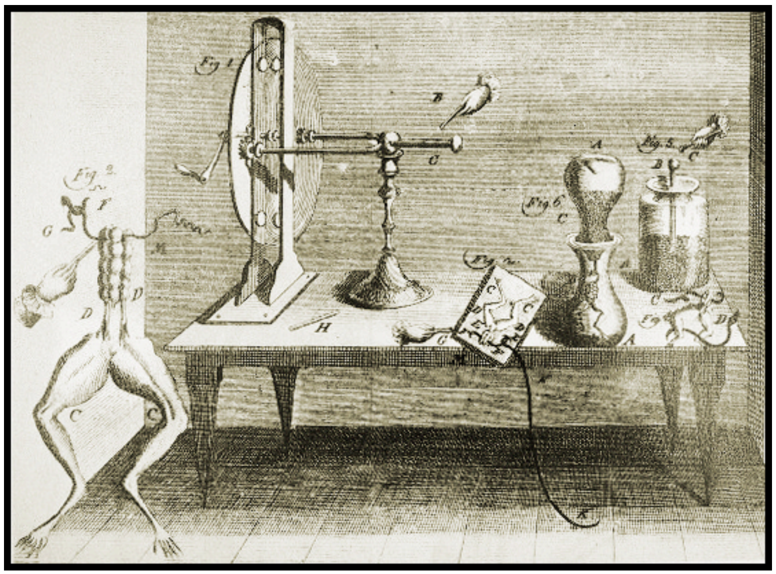
The Challenge

Electricity: 1800…



Electricity: Today…

$$\oint \vec{E} \cdot d\vec{A} = \frac{q}{\varepsilon_0}$$

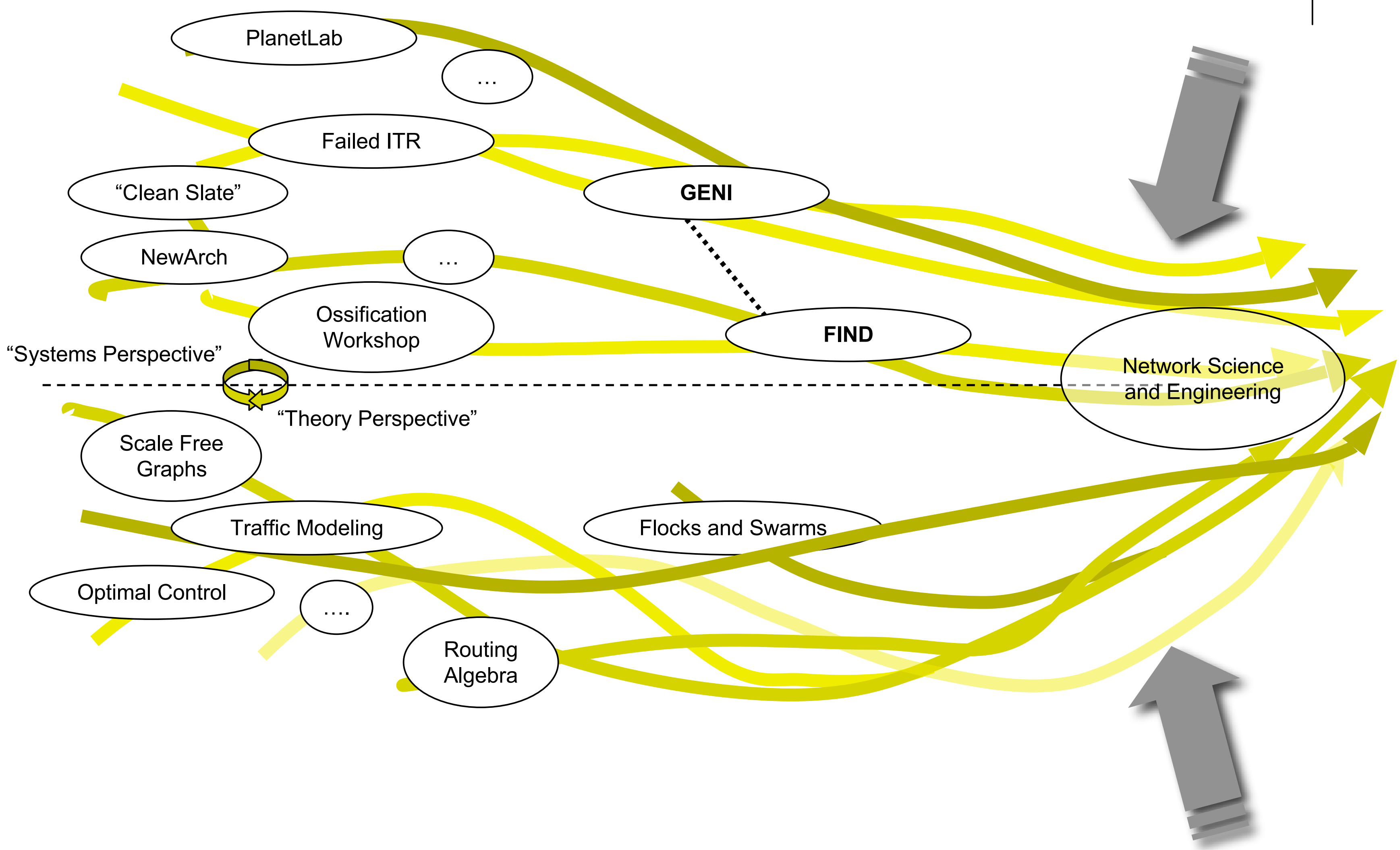
$$\oint \vec{B} \cdot d\vec{A} = 0$$

$$\oint \vec{E} \cdot d\vec{s} = -\frac{d\Phi_B}{dt}$$

$$\oint \vec{B} \cdot d\vec{s} = \mu_0 i + \frac{1}{c^2} \frac{\partial}{\partial t} \int \vec{E} \cdot d\vec{A}$$

Can we do the same
…for Network Architecture and Design?

Broad communities, many roots, different perspectives



Day

- This talk
  - FIND, NetSE, and (a touch of) GENI
    A systems-rooted perspective on future network architecture *-or-*
    where math and theory get complex..
- John Doyle
  - Architecture and theory
- Slightly structured discussion on the above..

FIND:
A systems perspective on Future Internet Architecture and requirements

This material is originally by David Clark, MIT CSAIL and FIND advisor to NSF.

Edits, simplifications, and opinions added..

The point of this segment is to review some things the systems / architecture folks are thinking about…

…and how they / we think..

…and where there is room to bring in rigor

Background

- FIND (Future Internet Design) is a U.S. NSF program to look at what our global network of 15 years from now should be.
- Rooted in the “systems / internet / networking” community.
- Similar efforts in Asia and Europe.
- U.S. program now about 3 years old.

FIND Outline

- Look at some of the important objectives
  - Why is this worth doing?
  - What is wrong with the network of today?
- Position the effort: FIND is both about:
  - The "packet" layer
  - Higher layers, closer to the users.
    - Design patterns for applications.
    - Application support services.
- Describe some emerging proposals and approaches
  - Sometimes conflicting, sometimes clear.
  - (Sometimes a personal point of view.)

The Internet is a success

- So why would we want to rethink its design?
  - It’s *not* the data plane.
  - Packets / IP have proven their generality, and we have polished the data forwarding function for years.
- It is *not* that some broad class of application is unsupported.
  - Application designers have shown the broad utility of the Internet.
- The issues are centered in the broader context within which the Internet is positioned.

Issues to consider

- Security
- Availability and resilience
- Better management
- Economic viability
- Meet society's needs
- Support for tomorrow's computing
- Exploit tomorrow's networking technologies
- Support tomorrow's applications
- Fit for purpose (it works…)

What are these??

- These are not requirements.
- They are a wish list.
  - Desiderata
  - An aide-memoire
- It is a big jump from any of these items to the design of mechanism.
  - And <u>that</u> is a big issue.
  - Can we (strive to) address it (better)?

Design methodology

- We must think about the process of moving from objectives to specific requirements to mechanisms and architecture.
  - If the problem is to big to consider at once, must modularize the design process.
    - How? Beware an over-dependence on layering.
    - What guides us, then?
- That list of issues represents a broad set of criteria:
  - Not just the “traditional”: performance/optimization, generality, new technology
  - Implies a multi-dimensional assessment of new ideas. Implies tradeoff and balancing.
- We understand a lot more now than we did in 1974.
  - This current work should be based on methodical design, analysis, theory
  - But in <u>many</u> places, we don’t know how yet…

Security

- Use as a first example.
  - Hard and important
- Why is the problem so hard?
  - We don’t agree on the definition of good security
    - No metrics (more on this later..)
    - A balance (or “tussle”) among stake-holders
  - We want different outcomes in different contexts
  - We cannot correct the insecurity of end-nodes
- Old ideas: (good ideas, but not why we thought)
  - Disclosure, integrity, availability - a reasoning structure
  - How does this relate to firewalls, VPNs? (mechanisms)
    - After the fact--not a part of the network
    - Doesn’t quite fit the reasoning structure?

A different modularity

- Attacks on communication (the network)
  - Confidentiality and integrity addressed with encryption
  - Availability?? The central objective of networks
  - What else?
- Attacks on the host (the endpoints)
  - Infiltration (can lead to most anything, hence essentially uncharacterizable)
  - So either prevent infiltration or limit its consequences..
- Denial of service
  - A special case of availability

Availability

- First, as much as possible, make attacks on communication into failures of availability (factor the problem)
  - Limit the range of attacks and responses.
    - Address some classes of attack with a transformation mechanism
  - E.g.: Mechanism: wrap an end-to-end confirmation of identity around a connection.
    - Cleanly makes attacks on/by the network into an availability problem.
- Second, develop a theory of availability.
  - At a high level:
    - All critical resources must be supported in a rich, heterogeneous, diverse form.
    - It must be possible to detect and distinguish (to some degree) failures.
    - The point of detection must be able to invoke different resources.
      - In general, only the end-points can detect failures.
      - In general, the end-points must be able to respond somehow.

Examples of attacks

- Byzantine routing.
  - Mis-routing and/or dropping.
  - Only end-node can ultimately detect, so end-node must be able to request re-routing
    - Explicit
    - Implicit
  - Multi-homed end-nodes may be important
- DNS corruption (pharming)
  - No architectural support today to mitigate this
    - Design is redundant, but not in face of malice

End-to-end checks

- To turn misdirection attacks into “availability problems”, need a means to confirm with whom you are communicating.
  - An issue of identity and shared information.
    - What notion(s) of identity will be suitable?
- “You” means the end-nodes, but not just the human. If the end-node can be trusted, software can help.
  - Corrupted end-nodes are a central issue here.
  - Can a trusted third-party helper node help?

Network management

- Even less structured than security
  - No real consideration in original design
  - Today: not *network* management, but remote management of boxes
- Possible (common) decomposition:
  - Network planning and configuration
  - Fault isolation and resolution
- Does this framing actually decompose the problem?
  - Do we know the modules of management?
  - What would be an appropriate criteria for modularization?

New ideas:

- Critical interfaces:
  - Between layers to integrate application, network and technology
  - Between regions to allow cross-domain capabilities with <u>partial information</u>.
    - This interface is fundamental. It reflects reality.
    - Modeling? Of what? Organizations?
- Expression of end-user *intent*
  - Critical in solving automation problem
- Better tools for configuration and planning
  - Critical in solving availability problem.
- Default management automatic, just like dynamic host configuration
  - Planning vs fault resolution, again.

Observations

- Management has a lot to do with security and availability
  - These two areas are not “modules”.
  - Cannot (best not…) have a “security” or a “management” design sub-group.
  - So how to “modularize”?
- For both areas, the community has lots of great point-source ideas, but we lack an accepted (let alone rigorous) approach to architectural framework.

Region interconnection

- Old idea: BGP
- New ideas:
  - Interconnection of advanced services ("composition in the net")
  - Direct expression of business constraints
  - Fault localization and correction
  - Interconnection of traffic aggregates
  - Short-term markets for service
  - Security issues
    - Control of DDoS
    - Detection of corrupted or untrustworthy regions

Economics and industry structure

- Remember, protocol interfaces (e.g. architectural modularity) induce industry structure.
  - There is no fundamental reason why ISPs look the way they do
  - What's an ISP for, anyway?
    - Expertise…
    - Capital…
  - What if you don't need one or the other any more?

(Here's a way to say)

Some specific requirements

- ISPs must be able to model usage and demand sufficiently well to make investment decisions
- Users must be able to select among paths through the network that avoid failures
- The network design must allow users a degree of choice among providers so as to impose the discipline of competition

- Note: avoided saying "routing"…

(A wrong way to say that…)

- ISPs must have control of routing to ensure that forwarding paths align with business arrangements
- Users must have control over routing to allow them to route around failures and improve performance
- Users must have control of routing to allow user choice and the discipline of competition

- Routing is a mechanism, not a requirement
  - In a future network, might do routing differently, and there would be no conflict…
  - These different ideas may be handled by different / decoupled mechanisms.
- (Where) can we usefully create “requirement formalism” (in contrast to “mechanism formalism”)?
  - Fundamental complexity / properties vs mechanism complexity

Managing investment

- ISPs must be able to model usage and demand sufficient to make investment decisions
- Conventional wisdom challenge:
  - Is it true that investment occurs on a longer time-scale and so must occur in advance of usage?
    - Obviously, for physical construction
    - But intermediate abstractions may defuse problem at higher layers
    - Virtualization or rapid reconfiguration

Long-term facilities investment

- Physical facilities (fibers, towers, etc.) require capital investment
  - How to justify investment?
- ISPs serve this function today
  - They don't just move packets, but manage capital and risk. Important economic role.
- To insulate facilities investors from risk so that they will invest, need futures market.
  - Happens today with really expensive cables.

A new interface

- Do we need to standardize the interface that defines this futures market?
  - Has a lot in common with other commodity markets
- Not sure, but if we do, it is an odd sort of standard
  - Not moving packets, but money
- Does *this* belong in an overall optimization framework?
- Modularity by timescale (a common theme..)

Information - moving up-layer

- Old idea: an application issue - ignore it
- New idea: need a framework
  - Naming and identity of *information*
    - Independent of how you get it
    - "Content based networking" - hot topic
    - But: think about privacy.
      - If you shout for information, the whole world hears.
      - Tradeoff, or orthogonal mechanism?
  - Dissemination
    - Swarms, P2P: (heterogeneous).

Design of Mechanisms

- The previous discussion (very incomplete) hints at the range of issues that designers of a future network should/will consider
- A future network will have mechanisms that (at a high level) are familiar, but they may take very different forms
- Can rigor / theory / math
  - Show the way?
  - Keep up at all?

Multiplexing--a basic issue

- Old (1960’s) idea: packets.
  - Seems to have worked out well.
- New ideas:
  - integrated management of packets and circuits (aggregates).
    - Integrated management.
    - Fault recovery, routing/traffic engineering.
    - Integrate future concepts in optics (routing vs. TE)
  - Virtualization of routers and links
    - Avoid need to have one design.
    - Needs assessment against our broad list of considerations.
    - And these two ideas need to be harmonized.

Addressing

- Old view: designed for efficient forwarding.
- New view: take into account
  - Security issues
    - Accountability, privacy, deterrence, hiding.
  - Management issues
  - Do you really want to address physical nodes?
    - How about services? Anycast?
    - But consider lower-layer management issues.
  - Multi-homing

Routing

- Old view:
  - Find the lowest cost route
  - Load-based dynamics lead to instability.

- New ideas:
  - Random route selection.
    - Avoids DoS on link
    - Avoids traffic engineering.
  - User route selection
  - Multi-path routing.
  - Machine learning to achieve high-level policies
  - Move route computation out of forwarders.
  - Multiple simultaneous routing schemes.
  - Routing regions that do not match facilities ownership.

Connection establishment

- Old idea: minimize the round trips.
- New ideas:
  - Need a phase for exchange of identity.
    - May need a "cross-layer" initial exchange
    - Re-modularize TCP to be less layered
  - Need to diffuse attacks
  - Adding a round trip or two (esp. if not always) worth the cost in order to allow an E2E check
    - Part of availability framework.
  - Fit this thinking into the DTN paradigm…

Application design

- Old view (simplistic): our machines talk.
- New view:
  - Lots of servers and services.
  - Need for cross-application core services.
    - Identity management, social networks.
  - Modulate behavior based on trust.
- Application design patterns and building blocks should be an *integrated* part of the future network.
  - Or not. Can we both specialize *and* generalize?

And more…

- Naming (of all sorts of things).
- Social context
- Other aspects of security (e.g. protecting the host)
- Computing and network technology
- …

Observations

- Mechanism (e.g. routing) is a response to a set of requirements, not a given.
  - Derive mechanism, don't presume it.
  - This is an excellent opportunity for our challenge!
- The (new) interesting interfaces will not involve packets but control, investment, social context, etc.
- Standardize as little as possible, but no less.

NetSE

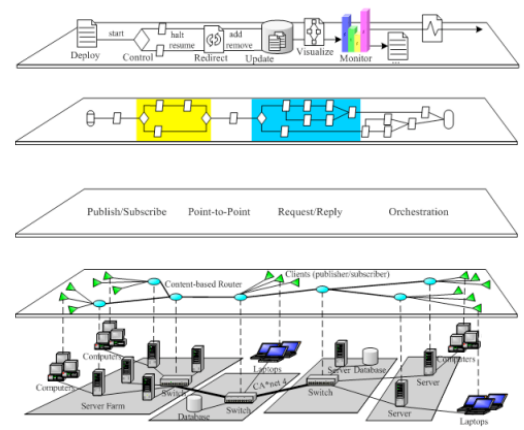
"Network Science and Engineering"

NetSE

- Starting in late 2007, US NSF and the Computing Community Consortium began to consider a new research agenda: “Network Science and Engineering”
- This addressed two distinct goals
  - Deal with a small anomaly in the development of GENI
  - Broaden the FIND-ish research agenda beyond systems

NetSE Challenge to the Community

- Fundamental Question: Is there a science for understanding the complexity of our networks such that we can engineer them to have predictable behavior?



Credit Middleware Systems Research Group

Slide credit: Jeannette Wing (edited..)

Network Science and Engineering: Fundamental Drivers

Society → Enable new applications and new economies, while ensuring security and privacy → **Security, privacy, economics, AI, social science researchers**

- Design secure, survivable, persistent systems, especially when under attack
- Understand technical, economic and legal design trade-offs, enable privacy protection
- Explore AI-inspired and game-theoretic paradigms for resource and performance optimization

Technology → Develop new architectures, exploiting new substrates → **Distributed systems and substrate researchers**

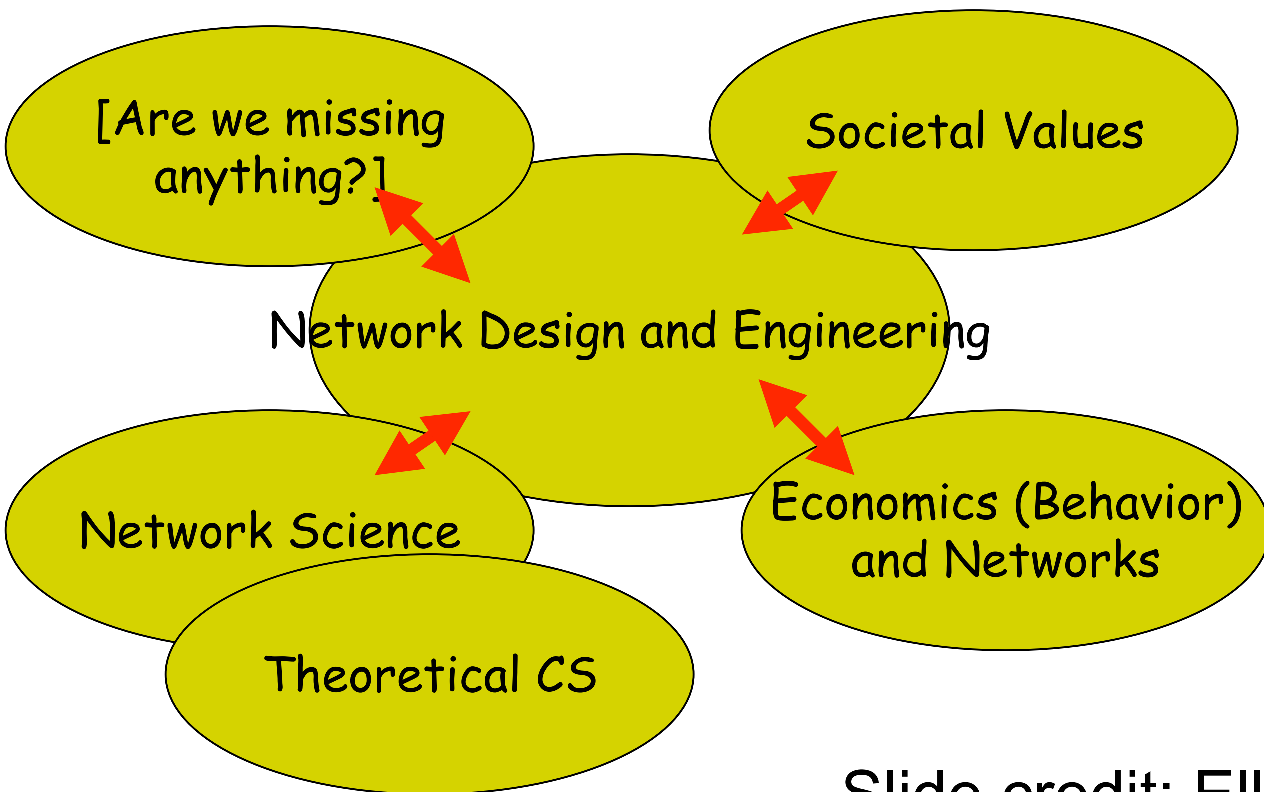
- Develop architectures for self-evolving, robust, manageable future networks
- Develop design principles for seamless mobility support
- Leverage optical and wireless substrates for reliability and performance
- Understand the fundamental potential and limitations of technology

Science → Understand the complexity of large-scale networks → **Network science and engineering researchers**

- Understand emergent behaviors, local-global interactions, system failures and/or degradations
- Develop models that accurately predict and control network behaviors

NetSE Planning Activity: Intellectual Space (not to scale)

Goal: Understand how to design, engineer and operate "better" networks



Slide credit: Ellen Zegura

CCC NetSE Timeline

- June-Sept 2008 – elaborate the space
  - workshops (3)
  - meetings (2)
- Oct 2008
  - draft research agenda completed
  - incl. recommendations on how to advance agenda
- Nov 2008
  - collect feedback (from few then many)
- December 2008
  - finalize research agenda

You are here
Agenda is a work in progress

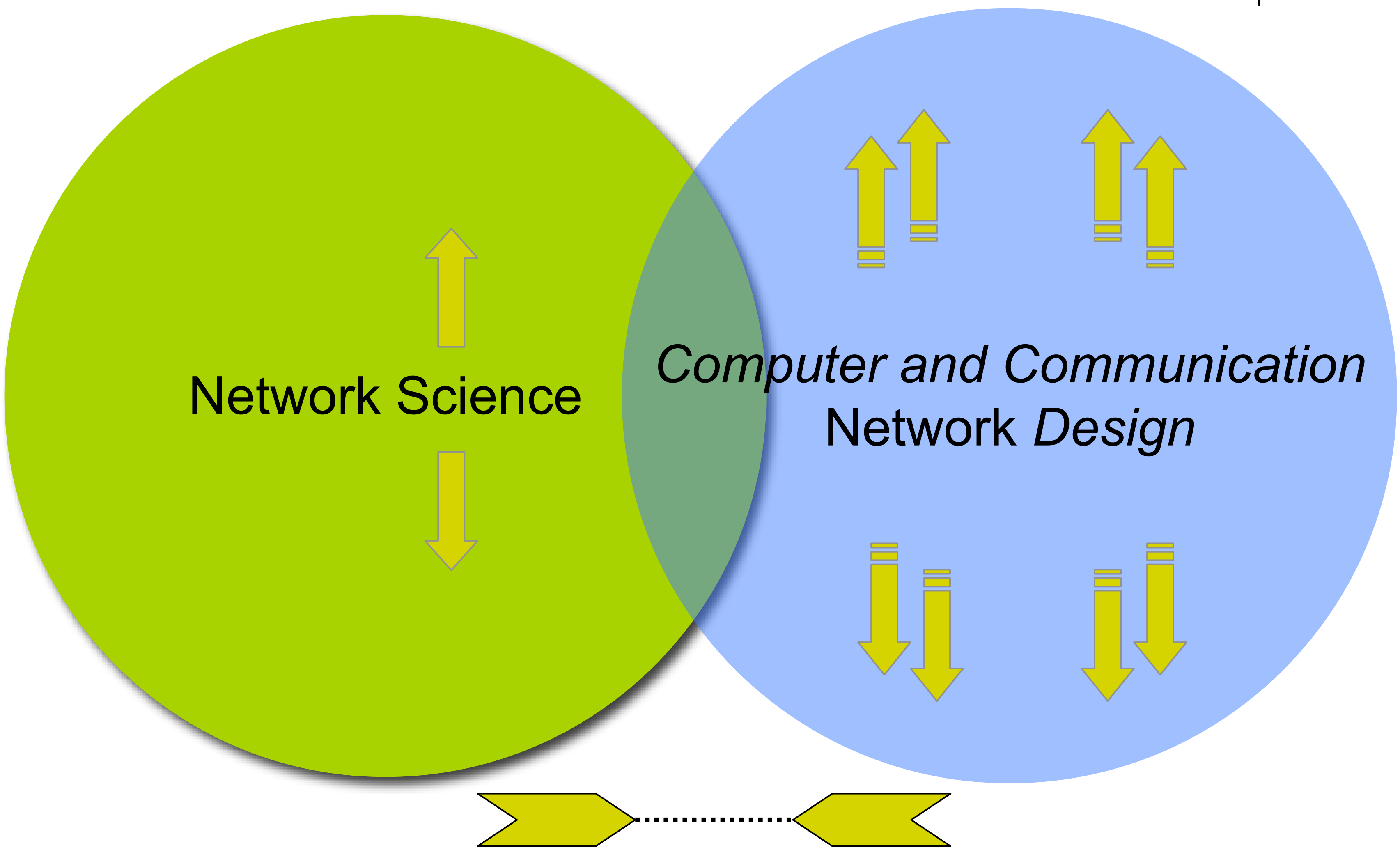
Workshops and Meetings

- Workshops
  - Network Science and Network Design
    - John Doyle, CalTech/NSF and John Wroclawski, ISI
    - July 29 and 30, Los Angeles
  - Behavior, Computation and Networks
    - Mike Kearns, U Penn and Colin Camerer, CalTech
    - July 31 and August 1, La Jolla
  - Network Design and Societal Values
    - David Clark, MIT and Helen Nissenbaum, NYU
    - September 24-26, Washington DC

- Meetings
  - Smaller than workshops
  - Extract/expand on more well trod areas
  - Theory and Network Design
    - John Byers (BU), Joan Feigenbaum (Yale), Ellen Zegura (GT)
    - June 11, Boston
  - Network Design and Engineering
    - Nick Feamster (GT), Amin Vahdat (UCSD), David Andersen (CMU), Mike Dahlin (UT Austin), Jen Rexford (Princeton), Craig Partridge (BBN), David Clark (MIT)
    - August 17, 18, Seattle at SIGCOMM

This material expands on a talk at the NSF NetSE Informational Meeting, September 2008:
"Report on NetSE Workshop on Network Science and Network Design, held July, 2008 at USC/ISI"

A key NetSE Subtopic: Connecting Network Science and Network Design

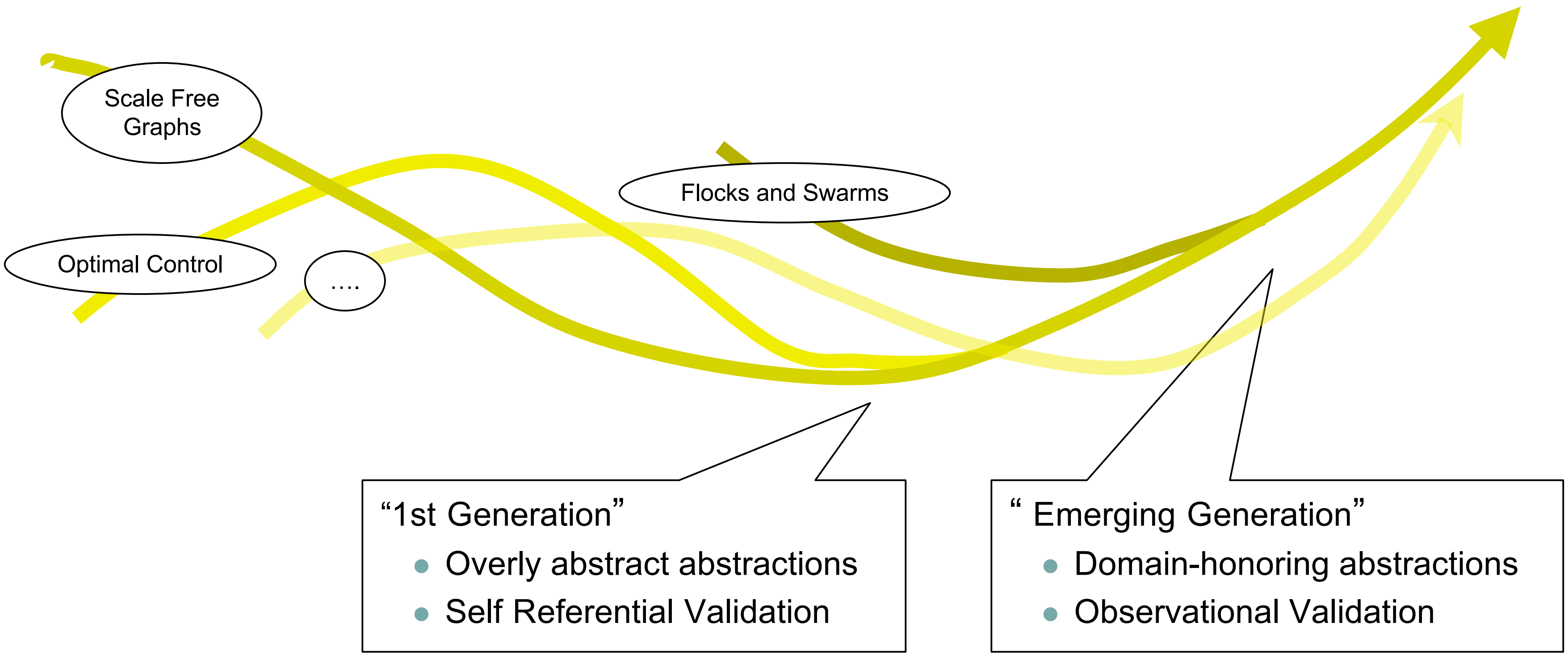


Views of Network Science

- Different people use this phrase in different ways
- An established community with its own culture and perspective
- Views:
  - “any theory that has to do with networks”
  - “power laws and scale free graphs”
  - Search for common abstractions, metrics, tools *across network domains*

Evolution…

- Search for common abstractions, metrics, tools *across network domains*
  - Powerful. Tricky.



The structure of scientific explanation

- Different sciences use levels unevenly and in very different ways.

Levels

1. Verbal
2. Data & stats
3. Modeling & sim
4. Analysis
5. Design & synth

- Network ***science*** has historically focused on data and statistics.
- Network ***design*** demands a fundamental rethinking / transition, particularly ***proofs*** in ***analysis and synthesis***.

This slide run: credit J. Doyle

Evolution of
Theory and the Internet

Goals
- Abstraction (common concepts across fields)
- Rigor (& math structure)

Issues
- Statics (topology, structure)
- Dynamics (location, propagation)
- Robustness (& security)

Levels of understanding
1. Verbal (& cartoons)
2. Data & statistics (Experiments & measurements)
3. Modeling & simulation
4. Analysis
5. Design & synthesis

Good news:
- Spectacular progress

Bad news:
- Persistent errors and confusion
- Potentially insurmountable obstacles?

“The Matrix”

- subfields of networking, and progress therein..

	Traffic	Topology
Verbal		
Data/stat		
Mod/sim		
Analysis		
Synthesis		

ARCHITECTURE

C&D	Layering	???

A success story
Traffic (1993-2000)

	Traffic
Verbal	
Data/stat	
Mod/sim	
Analysis	
Synthesis	

- Heavy tails (HT) in net traffic???
- Careful measurements
- Appropriate statistics
- Connecting traffic to application behavior
  HT files ⇒ HT traffic
- “optimal” web layout

A lesson learned
Topology (1999 - Present)

	Traffic	**Topology**
Verbal		
Data/stat		
Mod/sim		
Analysis		
Synthesis		

Control, Dynamics, and Architecture:

Exciting recent progress

	Traffic	Topology
Verbal		
Data/stat		
Mod/sim		
Analysis		
Synthesis		

ARCHITECTURE

C&D	Layering	???

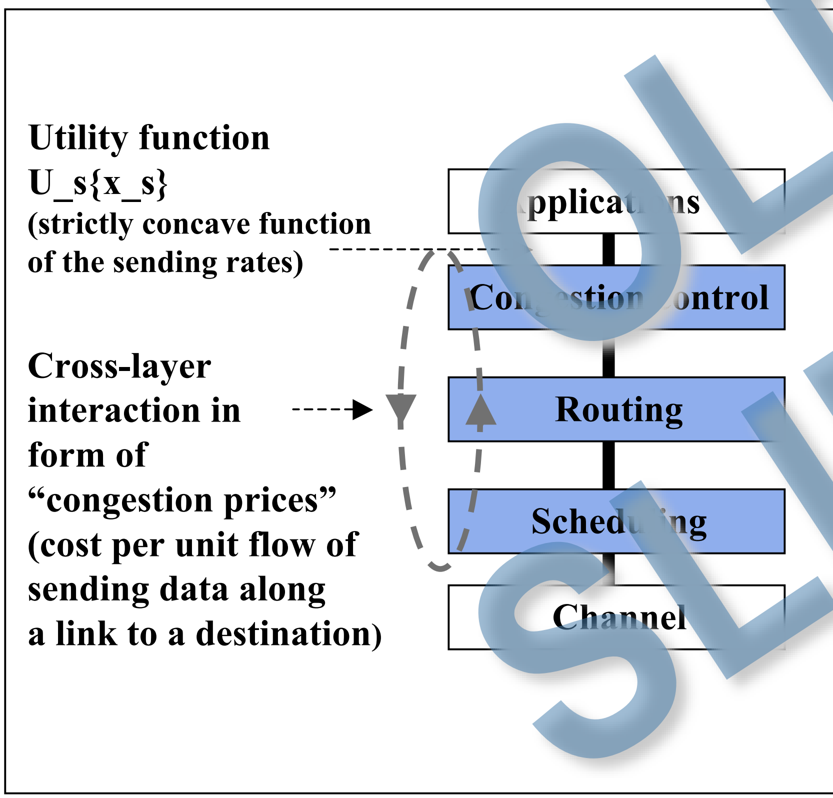
Three Research Examples

Extending a Theory

New Columns in the Matrix

Design by Constraint

Example: Extending the Theory
Theoretically Derived Architectures



- Network resource allocation formulated as global optimization problem
- Primal-dual decomposition generates a *set* of dual problems/algorithms/modules:
  - Local (except scheduling)
  - Tied together through congestion prices
- System <u>Architecture</u> traceable to theoretically provable optimality..

Optimal Cross-Layer Congestion Control, Routing, and Scheduling Design in Ad Hoc Wireless Networks. Lijun Chen, Steven H. Low, Mung Chiang†, John C. Doyle (Caltech and †Princeton)

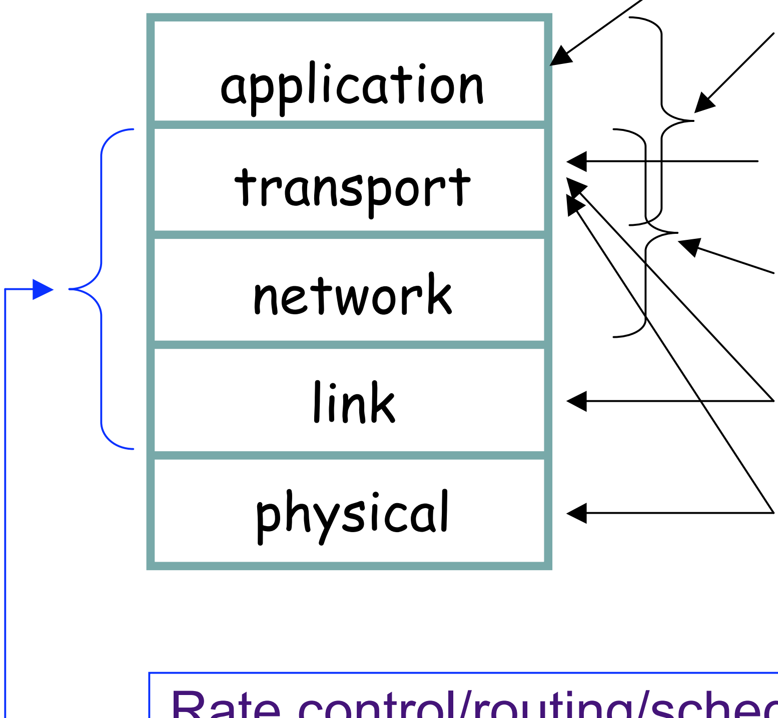
Example: Extending the Theory
New Challenge: Delay

- Previous work structured around *flow rate*
- *Delay* is the critical issue for many new applications:
  - Cyber Physical Systems (Networked Control)
  - Games, Interactive Communication, etc

- Approach: (attempt to) apply a tested methodology..
  - Enhance modeling to capture new effects (OK)
  - Identify and add new constraints to optimization problem (~OK)
  - Extend theory to operate in the presence of new constraints
    (So far, hard..)

- *Key result* if successful:
  Theoretically derived architecture for delay-sensitive networks

History: Continual Advance *through* Similar Methodology



Web layer: Zhu, Yu, Doyle '01

HTTP/TCP: Chang, Liu '04

TCP: Kelly, Maulloo, Tan '98, ...

TCP/IP: Wang et al '05, ...

TCP/MAC: Chen et al '05, ...

TCP/power control: Xiao et al '01, Chiang '04, ...

Rate control/routing/scheduling: Eryilmax et al '05, Lin et al '05, Neely, et al '05, Stolyar '05

Integrating network coding w/above: (Chen et al '07, Cui et al '07, ...)

Detailed Survey: Proc. of IEEE, 2007

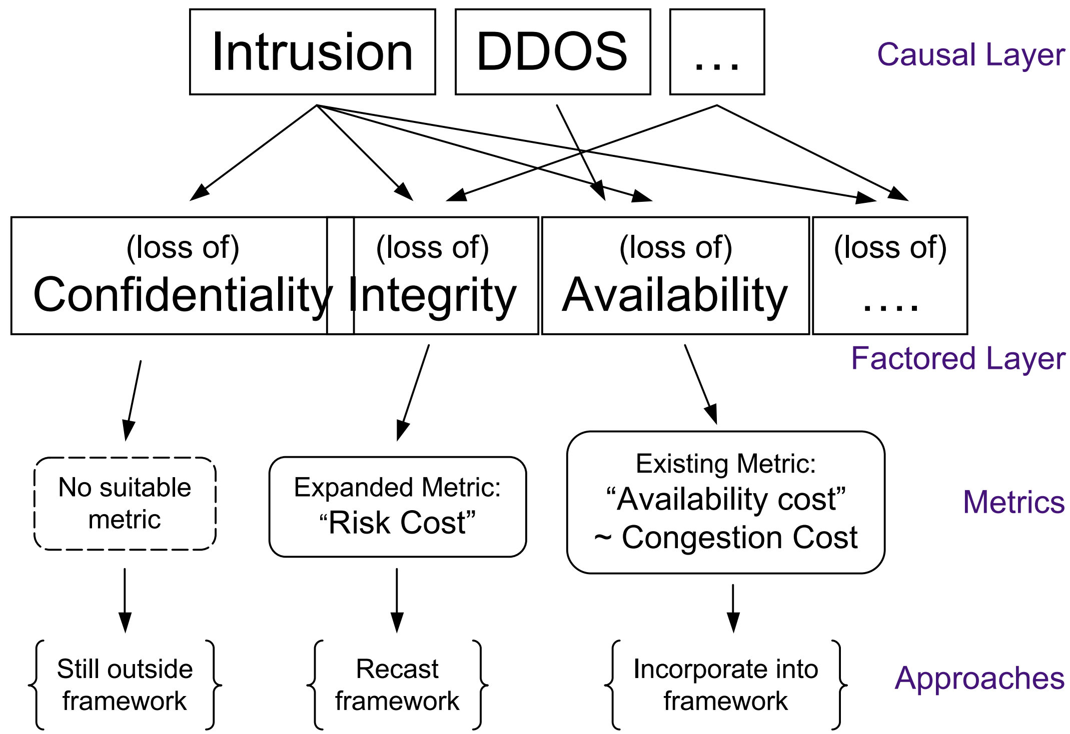
Example: New Columns in the Matrix “Security” in a theoretical framework?

Caveat: entire slide is insane speculation

(but note: close relation to FIND talk segment..)

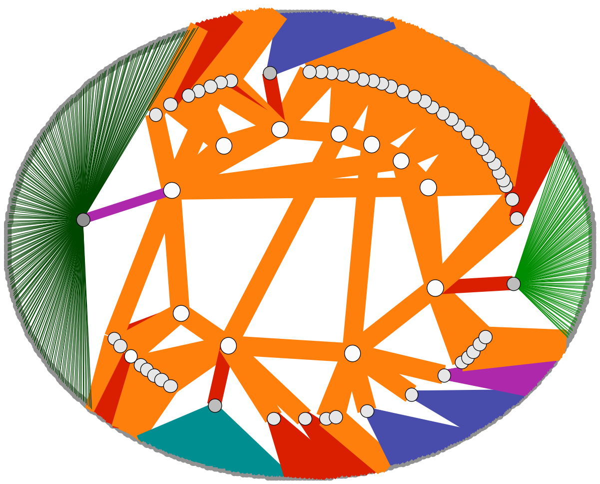
- Challenge:
  broaden theoretical frameworks to include additional design elements

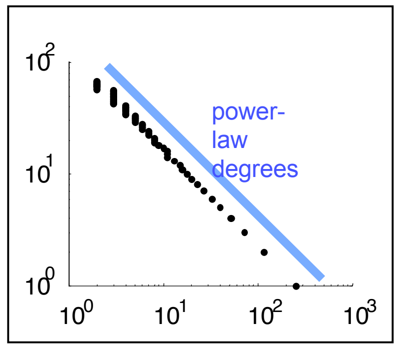
- Key issues:
  - Metrics
  - Relatable Metrics

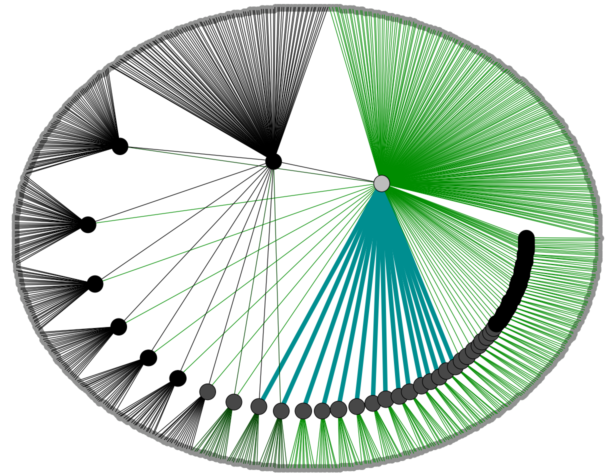


Example: Global Results from Local Actions
"Design by Constraints"

Two "Internet Topologies"; same power law parameters..







Low degree mesh-like core

- *High* performance and robustness
- Efficient, economic
- From "random" generator, *low* probability, but
- *Like* real Internet

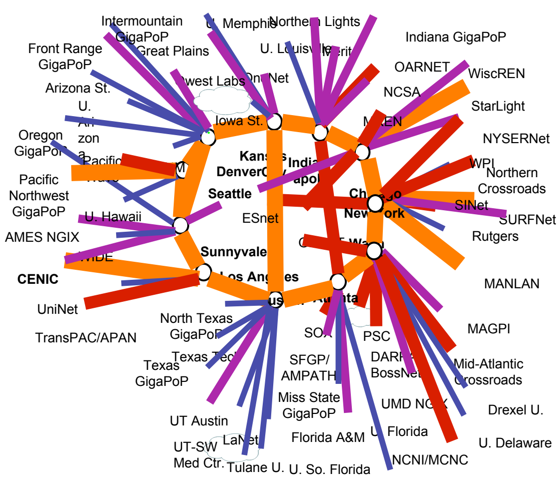
High degree hub-like core

- *Poor* performance and robustness
- Wasteful, expensive
- From "random" generator, *high* probability, but
- *Unlike* real Internet

What's going on?

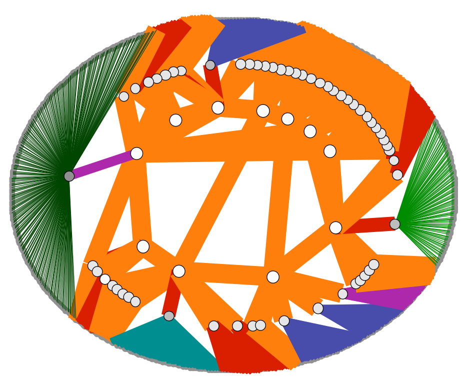
- This is *surprising* to many in network science
- This is *not surprising* to most Internet engineers
- What's going on?

Abilene Backbone Physical Connectivity



Start with an *engineered* backbone…

Add gateway routers and end systems *consistently with technological constraints* on these routers and systems…



Get topology [synthesized or real] with high throughput, efficiency, economy

Design by constraint

- The desirable topology is due to both
  - Classical engineering
  - *Local constraints* shaping *global results*
- To be fair, perhaps somewhat by accident..

- The key question: can we do it <u>on</u> <u>purpose</u>?
  - Design, not of the complete system, but of components from which systems with desired properties will come forth
  - Formalization of methods for this class of design

Validation

"I argue that power law research must move from focusing on observation, interpretation, and modeling of power law behavior to instead considering the challenging problems of validation of models and control of systems"

- Michael Mitzenmacher*

- *Validation* is essential to the progress of science and engineering
- We need some attention here…
- Three types:
  - "Self Referential Validation"
  - Observational Validation
  - Generative (design-based..) Validation

**Editorial: The Future of Power Law Research*.
M. Mitzenmacher, Internet Mathematics 2(4), 2006

Self-Referential Validation

- Identify a phenomenon in/of the original artifact
- Develop a mathematical model that captures that phenomenon..
- Note similarity.

- Key question: what is being validated here?

Observational Validation

- "Classic Science"
  - Model artifact based on observation of phenomena
  - Use model to predict *different* correlating/supporting phenomena
  - Observe artifact to validate model
- Key questions
  - Correlation between primary and validating phenomena
  - Observational platform capabilities (Internet)
    - Today, often focused on observing primary phenomena
    - Partial information and other observational problems

Generative Validation

- "Build it and see/study what happens"
  - For computer systems, often the most convincing approach
  - For standard engineering problems, no "fundamental" difficulty
- Key question: Type 3 problems*
  - Validating evolution over time
  - Validating the results of others' actions..
- The heart of our challenge, yet the hardest of all

*("global results from local action", "design by constraint", etc..)

Challenge: Validating *Architecture*

- This "Type 3" problem is closely related to the evaluation and validation of *architecture*
  - Architecture is the framework in which evolution and emergence occur
  - Architecture is all about fixed points (constraints)
- Can we as a community devise a more {effective, complete, rigorous} methodology for experimental architecture research?

Architecture (systems view)

To study and evaluate something, it is helpful to know what you are studying..

- "The high level principles that guide the technical design of a system, especially the engineering of its protocols and algorithms"
- Two distinct levels:
  - Structuring principles
  - Structure itself - functional decomposition and system modularity

Evaluation Criteria

- What does it mean for an architecture to be *good?*
  - Not a single, simple metric
  - Requirements are *broadly defined* and *hard to capture.*
  - *Multidimensional*, *conflicting*, requirements. Consideration of multiple tradeoffs…
  - Goals not evaluable (today) in quantifiable terms.
  - Goals concerned with behavior of system *over a long period of time.*

Good, redux

- What does it mean for an architecture to be *good?*
  - Suitability for purpose
    - Works..
  - Modularity of function
    - Distributed implementation, design, evolvable, etc - *multigenerational system lifetime*
  - Modularity and management of tussle[1]
    - Applicability and survival in the real world

[1] D. D. Clark, J. Wroclawski, K. R. Sollins and R. Braden Tussle in Cyberspace: Defining Tomorrow's Internet. IEEE/ACM Trans. Netw. 13, 3 (Jun. 2005), 462-475.

Basic Experimental Methodology

- Living the legacy of Descartes[1]
  - Formulate Experiment
  - Observe and evaluate Results
  - Rinse and Repeat

- Three key questions:
  - How to formulate intelligent architectural experiments?
  - How to execute experiments and evaluate results?
  - ~~What can I graph?~~

[1]René Descartes, *Discourse on the Method of Rightly Conducting the Reason in the Search for Truth in the Sciences,* 1637

Two parts to the story

- Architecture experiment *definition*
  - Devise experiments that challenge the system (worst case..)
  - Help identify places to look
- Architecture experiment *evaluation*
  - Reach accurate conclusions…
    …in a useful timeframe.

Experiment definition

- *Environment* must provide
  - Appropriate building blocks
  - Appropriate properties and “test jigs”
  - Composability
- Increasingly, formal tools can help
  - Real Options
  - Optimization Theory
  - Game Theory
  - Stakeholder Analysis
  - …

Strategies for Evaluation

- S1: Deployment and observation under realistic conditions
  - The “real users” argument
  - (not entirely satisfying..)
- S2: Accelerated Evolution
  - Speculate about world 10-15 years into future
  - Try to simulate conditions present at that time
    - Wider performance variation, shifts in usage, etc.
  - Other ways to implement the concept?

Strategies for Evaluation

- S3: Build on top of it
  - Evaluation through use.
    - By system component builders (eg, routers)
    - By system users (eg, applications)
- S4: Intentional Perturbation
  - Subject the system to a range of failures and stresses.
  - Measure and assess resilience and dynamics
  - “Active experiments” vs passive observation

Strategies for Evaluation

S5: Integration of experimental and analysis tools

- Tools such as DD-optimization, MD, etc may help
  - Identify likely worst case scenarios and situation for careful experimental evaluation
  - Establish theoretically grounded baseline performance bounds for comparison
  - Identify and quantify costs of complexity, communication
  - …

Strategies for Evaluation

S6: Stakeholder Analysis

- "A technique used to identify and assess the importance of key people, groups of people or institutions that may influence the success of your activity or project"
- Produces
  - List of key stakeholders
  - Interests and motivations
  - Ranking of relative importance of these factors
- Can help
  - Consider how best to emulate or model stakeholders most important to the experiment (fake real users?)
  - Perhaps, identify those safely ignored in simplified experiment
  - Analyze tussle characteristics of architecture

Challenge

- Can we as a community devise a more {effective, complete, rigorous} methodology for architecture evaluation and validation?
- This challenge is among the most difficult facing the architecture systems/theory field

GENI

"Global Environment for Network Innovation"

Key Points

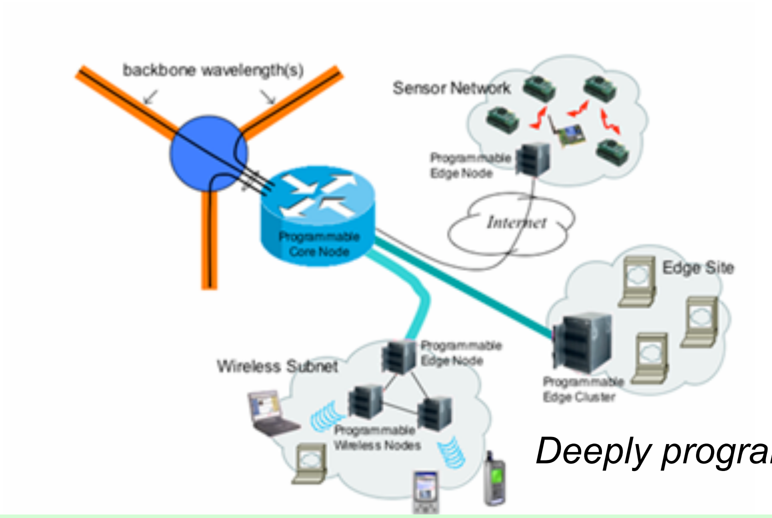
- GENI is an *infrastructure* effort
  - Like a telescope or a particle accelerator
- It has two purposes, weighted differently in different people's minds.
  - To further the validation task just discussed
  - To serve as a "deployment platform" for promising new ideas

The "official" summary

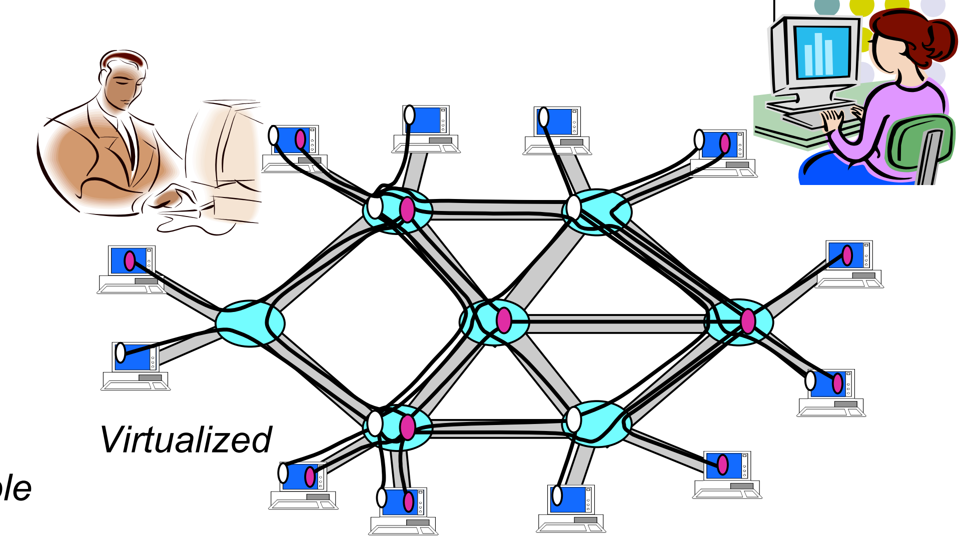
- GENI is meant to enable . . .
  - Trials of new architectures, which may or may not be compatible with today's Internet
  - Long-running, realistic experiments with enough instrumentation to provide real insights and data
  - 'Opt in' for real users into long-running experiments
  - Large-scale growth for successful experiments, so good ideas can be shaken down at scale
- A reminder . . .
  - GENI itself is <u>not</u> an experiment !
  - GENI is a suite of facilities on which experiments run

Slide Credit: GENI Program Office

Key GENI Design Elements

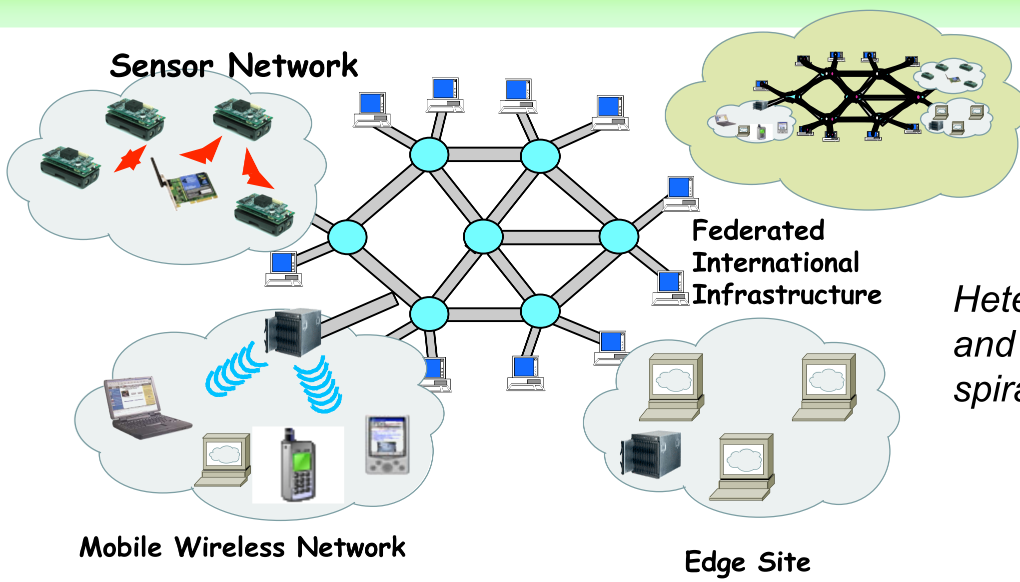


Deeply programmable



Virtualized

Programmable & federated, with end-to-end virtualized "slices"

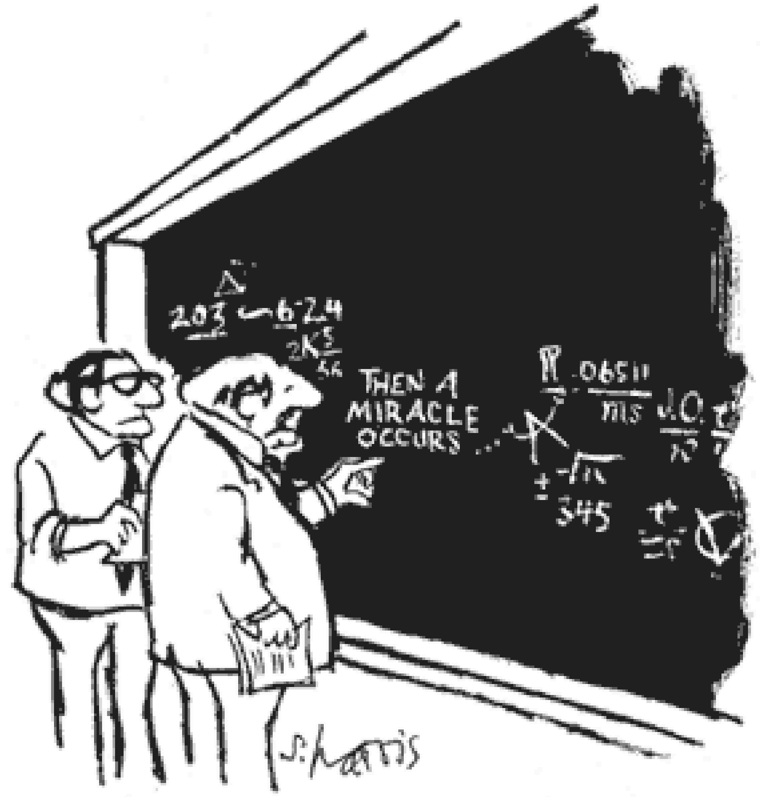


Heterogeneous,
and evolving over time via
spiral development

Questions

- Is this transformational, or anti-transformational?
- How can GENI best support the validation challenge outlined above?
  - Measurement
  - Environment and tools
  - “Scientific workflow”
  - …
- Right now, most focus on the basic prototyping infrastructure. Input from the arch/theory community is very much needed.

Integrating Network Science and Network Design



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

"Just a little bit further to go.."