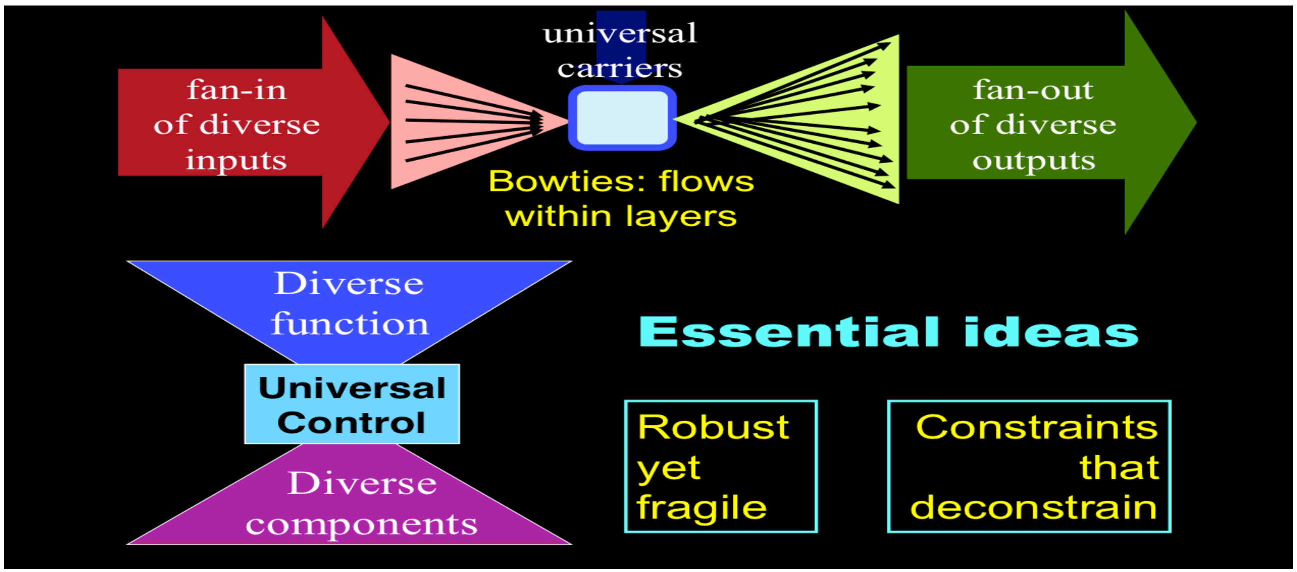
Macro Trends, SDN, and the Hidden Nature of Complexity



David Meyer
CTO and Chief Scientist, Brocade
Director, Advanced Technology Center, University of Oregon
SDN and Openflow World Congress 2013
Bad Homburg, Germany
dmm@{brocade.com,uoregon.edu,1-4-5.net,…}

Agenda

- Introduction – What is this talk all about?
  – New ways to think about all of this
- Macro Trends
  – Terminology: $SD^{(a/e)}$ → Software Defined Anything/Everything
- Complexity and its Hidden Nature
  – And its effects on/interaction with Cloud, NFV, and $SD^{(a/e)}$
- Summary and Q&A if we have time

Danger Will Robinson!!!

This talk is intended to be controversial/provocative (and a bit "sciencey")

But First….

- A few quotes from *Network Virtualisation – An Opportunity to Build Mouldable Networks, Telefonica, 2013*
  - *"Business development requires a continuous **evolution** of our network."*
  - *"…flexibility becomes the only way to efficiently adapt to our customer's needs in a scenario with increasing **uncertainty.**"*
- Interestingly, evolution and uncertainty are also biological concepts
- So…
  - *Networks must be **evolvable***
    - What architectural constructs create network evolvability?
  - ***Uncertainty** is inevitable, learn to live with it*
    - BTW, is it possible to *benefit* from uncertainty?
    - Hint: yes

But First, cont…

- As we will see, ***evolvability*** (along with scalability) is a key property we seek from Cloud/NFV/SDN

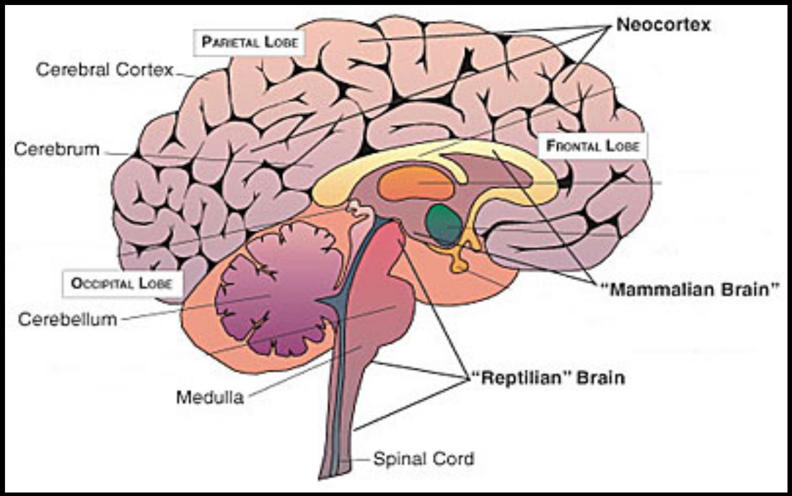
- OTOH*,* ***uncertainty*** turns out to be an inherent property of any sufficiently advanced technological system (or biological system for that matter), and shapes architecture and complexity. In particular
  - *"…complexity in highly organized systems arises primarily from design strategies intended to create* ***robustness*** *to* ***uncertainty*** *in their environments and component parts."*

- Systems have components and operating environments seem almost graciously uncertain, yet we (and biology) build ***robust*** systems.
  - How is this possible?

Let's Take A Look At A Few Macro Trends

Trend: The Evolution of Intelligence

Precambrian (Reptilian) Brain to Neocortex → Hardware to Software

HARDWARE	SOFTWARE
	

The rise of Cloud, SD$^{(a/e)}$ and NFV + Moore's Law

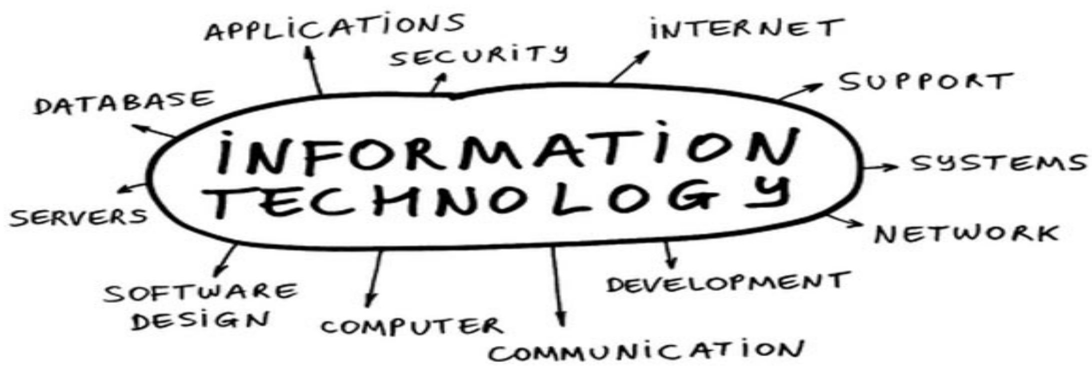
Note biological metaphor(s) – this is neither accidental nor superficial

Put Another Way

Trend: Everything De-silos

- Vertical -> Horizontal Integration
- Everything Open {APIs, Protocols, Source}
- Everything Modular/Pluggable
- ***Future is about open, horizontally scaled ecosystems***
    - Interestingly this too has a biological analogy: HGT

Trend: Network/IT "Fusion"



- Shift in influence, speed, and locus of purchasing influence
- Changes in cost structures
- **NetOPs + DevOPs ≈ Networking + IT ≈ Cloud**
- **SD$^{(a/e)}$ and NFV → Speed and agility → Cloud**
- **SD$^{(a/e)}$ and NFV also ≈ Network Virtualization**
- **Growing need for explicit notion of "service models"**

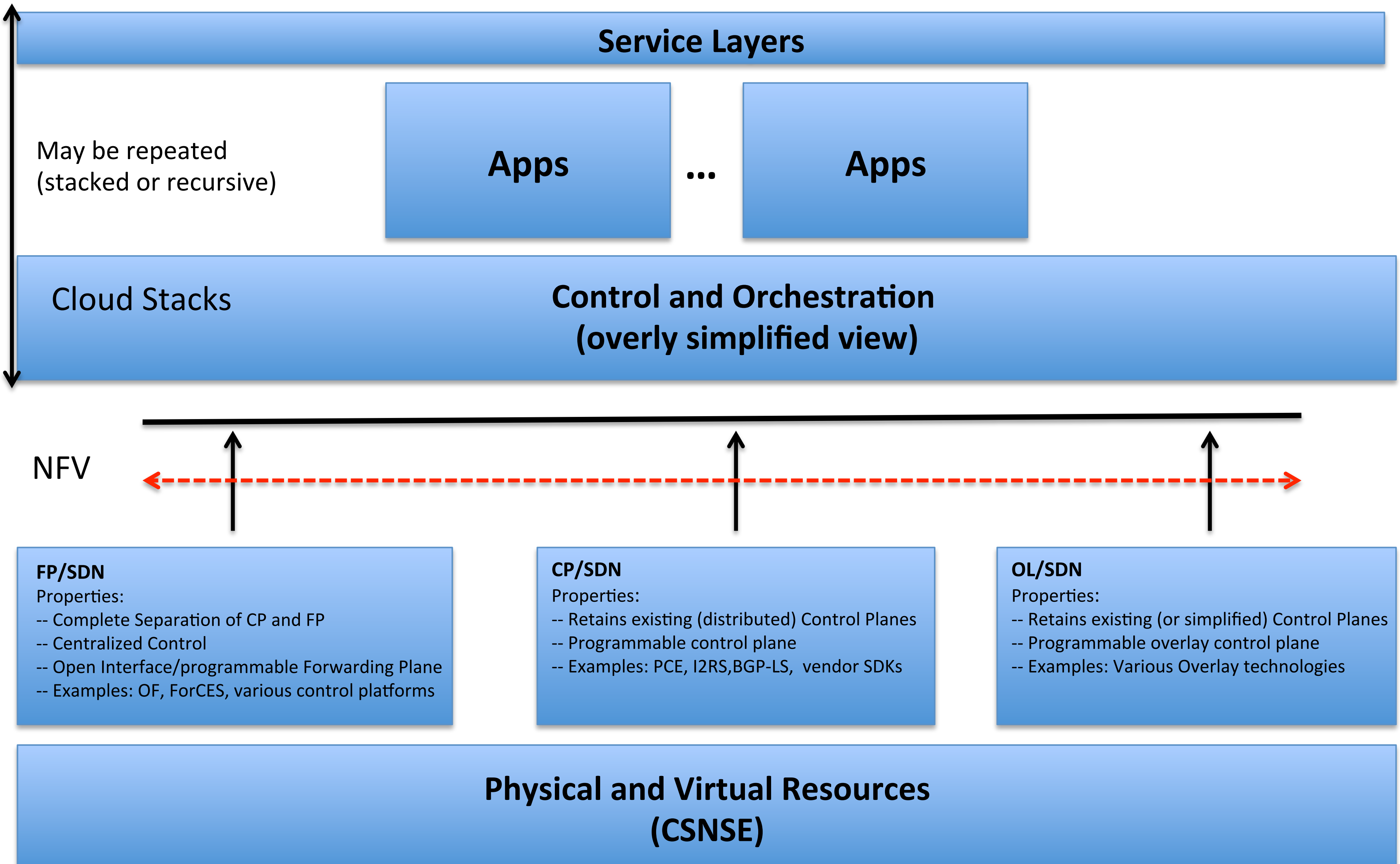
Other Important Macro Trends

- Everything Virtualizes
  - We've seen this…
  - Well maybe: Linux Containers, micro servers, …
- Data Center new "center" of the universe
  - Looks like ~ 40% of all traffic is currently sourced/sinked in a DC
  - Dominant service delivery point
- Integrated orchestration of almost everything
  - Automation based on cloud orchestration
  - Cloud orchestration, NFV, and SD$^{(a/e)}$ emerging as key architectural elements
  - "NFV seems to be the only movement toward orchestrating the cloud/network fusion that has any realistic chance of success ", http://blog.cimicorp.com/?p=1489
- Bottom Line: Increasing influence of software *everywhere*
  - Integrated orchestration of CSNSE++;
  - Note: increasing open loop control + s/w + Moore's law → increasing volatility/uncertainty

A Perhaps Controversial View on $SD^{(a/e)}$

- OF/SDN is a point in a larger design space
- The larger space includes
  - Control plane programmability
  - Orchestration of all kinds
  - Various architectural models including overlays
  - Compute, Storage, Network, Security, Energy (CSNSE)
  - $SD^{(a/e)}$ and NFV "convergence"
- My model: "$SD^{(a/e)}$ continuum"

A Simplified View of the *$SD^{(a/e)}$ Continuum*



So what are Robustness and Fragility?

- ***Definition***: A *[property*] of a *[system]* is **robust** if it is *[invariant]* with respect to a *[set of perturbations]*
  - Robustness is the preservation of a certain property in the presence of *uncertainty* in components and/or the environment

- **Fragility** is the opposite of robustness
  - If you're fragile you depend on 2nd order effects (acceleration) and the "harm" curve is concave
  - A little more on this later…

- A system can have a *property* that is *robust* to one set of perturbations and yet *fragile* for a *different property* and/or perturbation → the system is **Robust Yet Fragile (RYF complex)**
  - RYF complexity is a key property of biological and advanced technological complex systems
  - Characterized by Complexity/Robustness spirals

- Example: A possible ***RYF tradeoff*** is that a system with high efficiency (i.e., using minimal system resources) might be unreliable (i.e., fragile to component failure) or hard to evolve
  - Example: VRRP provides robustness to failure of a router/interface, but introduces fragilities in the protocol/implementation

- The *RYF tradeoff* is a hard limit and RYF behavior is "conserved"
  - You can't get rid of RYF behavior in advanced technological systems

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

System Properties Cast as Robustness

- **Scalability** is robustness to changes to the size and complexity of a system as a whole
- **Evolvability** is robustness of lineages to changes on *long* time scales
- Other system features cast as robustness
  - **Reliability** is robustness to component failures
  - **Efficiency** is robustness to resource scarcity
  - **Modularity** is robustness to component rearrangements
- So robustness is a key property we will want to consider
  - More on all of this in a few slides

RYF Examples

Robust	Yet Fragile
☺ Efficient, flexible metabolism	☹ Obesity and diabetes
☺ Complex development	☹ Rich microbe ecosystem
☺ Immune systems	☹ Inflammation, Auto-Im.
☺ Regeneration & renewal	☹ Cancer
Complex societies	☠ Epidemics, war, …
Advanced technologies	💣 Catastrophic failures

- "Evolved" mechanisms for robustness *allow for,* even *facilitate,* novel, severe fragilities elsewhere

- Often involving hijacking/exploiting the same mechanism that provides robustness
  - We've certainly seen this in the Internet space
  - Consider DDOS or other amplification attacks of various varieties

Brief Aside: Fragility and Scaling
(geeking out for a sec…)

- A bit of a formal description of fragility
  - Let *z* be some stress level, *p* some property, and
  - Let *H(p,z)* be the (negative valued) harm function
  - Then for the fragile the following must hold
    - ***$H(p,nz) < nH(p,z)$ for $0 < nz < K$***

- For example, a coffee cup on a table suffers *non-linearly* more from large deviations (*H(p, nz)*) than from the cumulative effect of smaller events (*nH(p,z)*)
  - So the cup is damaged far more by *tail events* than those within a few σ of the mean
  - Too theoretical? Perhaps, but consider: ARP storms, micro-loops, congestion collapse, AS 7007, …
  - BTW, nature requires this property
    - Consider: jump off something 1 foot high 30 times v/s jumping off something 30 feet high once

- When we say something scales like $O(n^2)$, what we mean is the damage to the network has constant acceleration (2) for *weird* enough n (e.g., outside say, 10 σ)
  - Again, ARP storms, congestion collapse, AS 7007, DDOS, … → non-linear damage

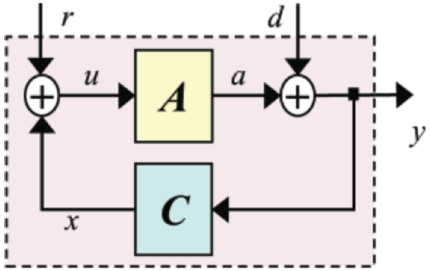
So What is *Complexity*?

"In our view, however, complexity is most succinctly discussed in terms of functionality and its robustness. Specifically, ***we argue that complexity in highly organized systems arises primarily from design strategies intended to create robustness to uncertainty in their environments and component parts.***"

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

Sound Familiar?

- Trend is towards designing for failure
  - Already the case in hyper-scale data centers
    - Design for robustness to uncertainty in the environment and component parts
  - This design strategy is exactly what gives rise to what we call "complexity"



- We use (s/w) control planes to overcome uncertainty
  - Consider original Internet design objectives
  - We frequently use feedback control (control planes)
- Why Feedback Control?
  - Because is relatively easy to build either uncertain, high-gain components or precise, low-gain ones; but the precise, high-gain systems essential to both biology and technology are impossible or prohibitively expensive to build unless a feedback control strategy is used
  - e.g., TCP (robust [risk aware] feedback controller)
- Not surprisingly, both technology and biology also use types other of control circuits
  - Consider low-pass filters: ISIS interface hold-down timer v. E.Coli C1FFL-AND "motif"
    - Responds only to persistent signals/filters spurious inputs
  - An Incoherent FFL is a pulse generator (multi-wave pattern)
    - But can also provide relative changes in signal (Webber's Law); factor away the ambient signal

See S. Mangan, A. Zaslaver and U. Alon, "The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks", http://ender.bu.edu/~tgardner/be567_readings/Cell signalling/Feedforward/Mangan.S_JMB_03.pdf

BTW, This Might Also Obvious But…

- Networks are incredibly general and expressive structures: *G = (V,E)*

- Networks are extremely common in nature
  - Gene expression, immune systems, energy metabolism, transportation systems, health care systems, legal systems, *Internet*, macro economies, forest ecology, the main sequence (stellar evolution), galactic structures, ….
  - Network Motifs
    - Motifs differ between transcriptional networks, food webs, social networks, neural networks, …

- So it comes as no surprise that we study biological and technological systems in our attempt to gain a deeper understanding of complexity and the architectural features that provide for scalability, evolvability, and the like.

- Why do we find the same universal architectural features in both biology and advanced technology?
  - ***Theory: Convergent Evolution***
  - Consider flight: If you want to fly wings turn out to be a good idea

- Ok, this is cool, but what are the key architectural takeaways from this work for us ?
  - where *us \in {ops, engineering, architects …}*
  - And how might this effect the way we build and operate CSNSE networks?
  - Keep this question in mind…

Ok, Key Architectural Takeaways?

- What we have learned is that there are ***fundamental architectural building blocks and associated behaviors*** that are found in systems that scale and are evolvable

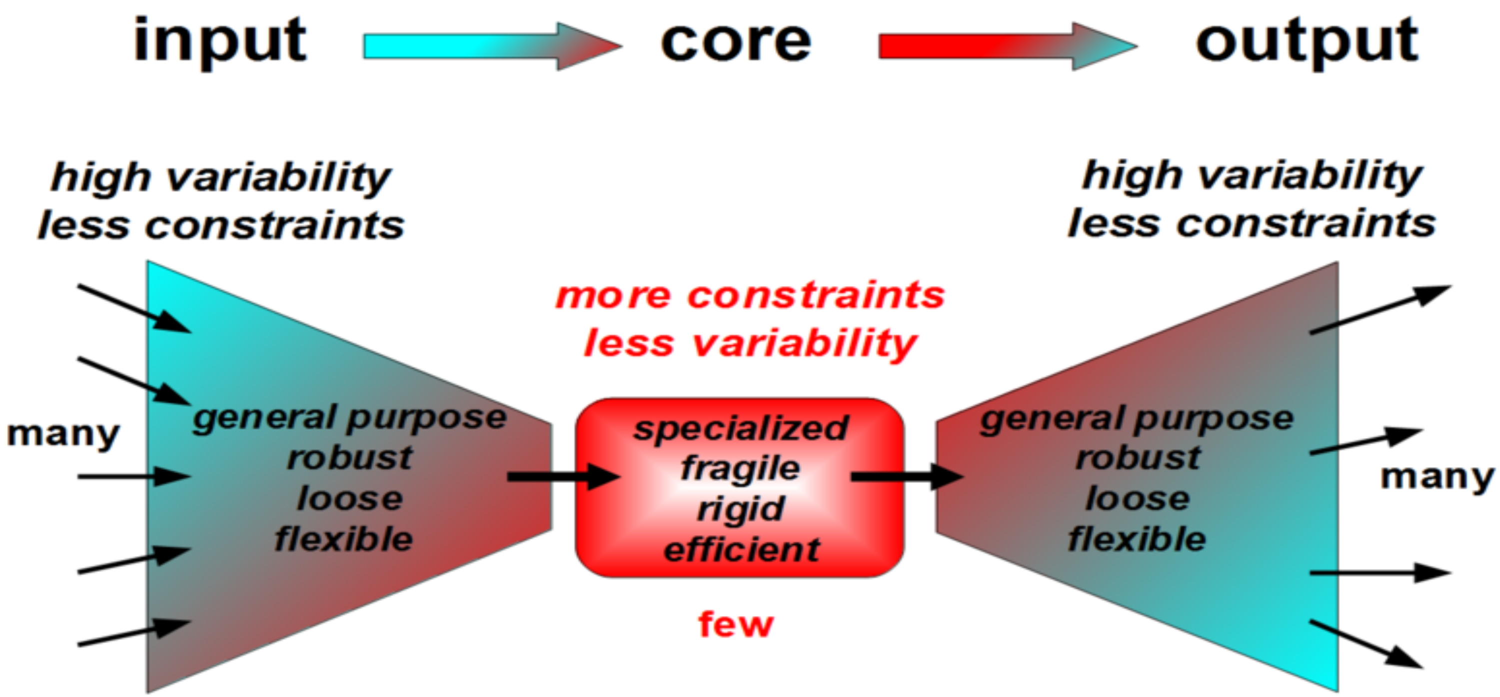
- The Hidden Nature of Complexity – RYF Complexity
  - RYF complexity is found in all biological and advanced technological systems and remains largely hidden
  - Reveals itself only during rare catastrophic, cascading failures [0]
  - Importantly, note that "These puzzling and paradoxical features are neither accidental nor artificial, but derive from a deep and *necessary* interplay between complexity and robustness, modularity, feedback and fragility." [1]

- Architectural Features
  - **Bowtie architectures**
    - We'll talk more about this in a few slides
  - Protocol Based Architectures (PBAs)
    - Fundamental to modularity; consider TCP/IP "plug and play"
  - Massively distributed with *robust* (risk aware) feedback control loops
    - Contrast optimal control loops and hop-by-hop control
  - Highly layered
    - But with layer violations, e.g., Internet, overlay virtualization
  - Degeneracy
  - Multi-scale Resolution (for example, time scales)

[0] "Any sufficiently advanced technology is indistinguishable from magic" -- Arthur C. Clark
[1] Csete, M. and J. Doyle, "Reverse Engineering of Biological Complexity", http://www.sciencemag.org/content/295/5560/1664.abstract

Bowties 101

Constraints that Deconstrain

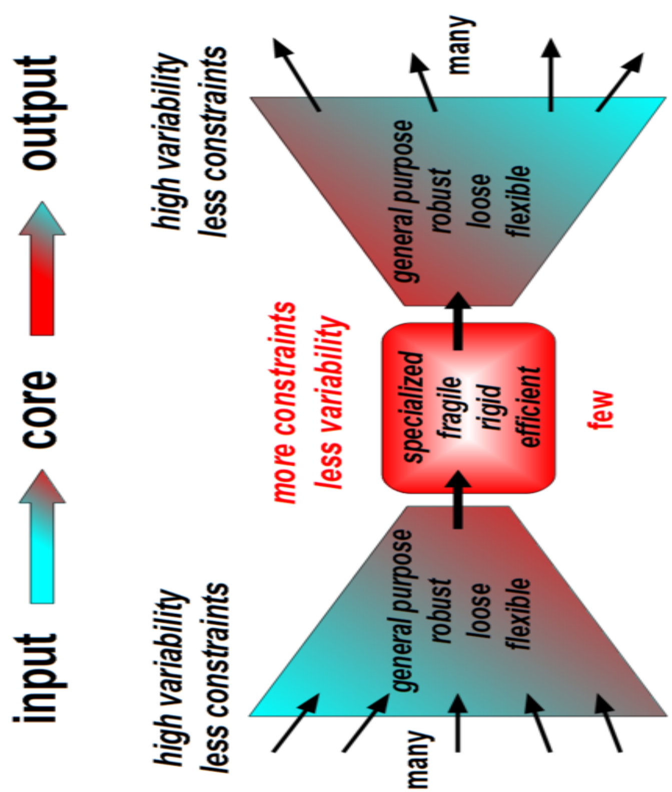


For example, the reactions and metabolites of core metabolism, e.g., *ATP metabolism*, Krebs/Citric Acid cycle signaling networks, …

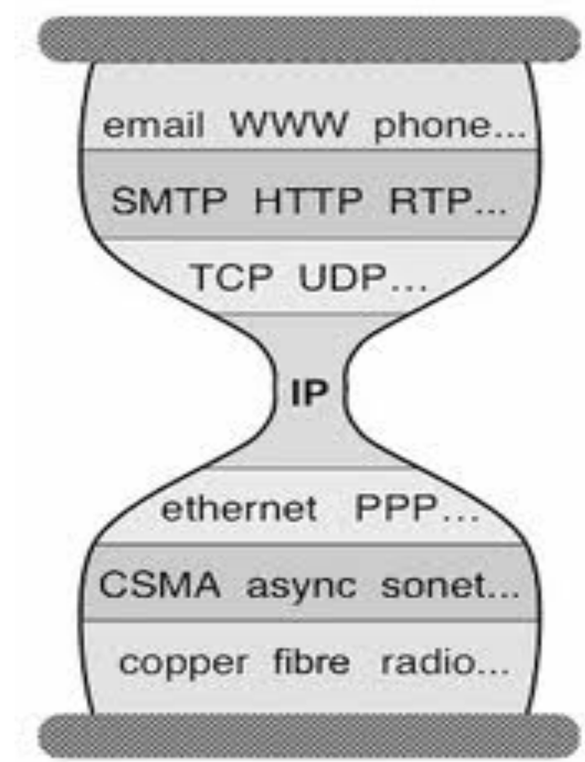
See Kirschner M., and Gerhart J., "Evolvability", Proc Natl Acad Sci USA , 95:8420–8427, 1998.

But Wait a Second

Anything Look Familiar?



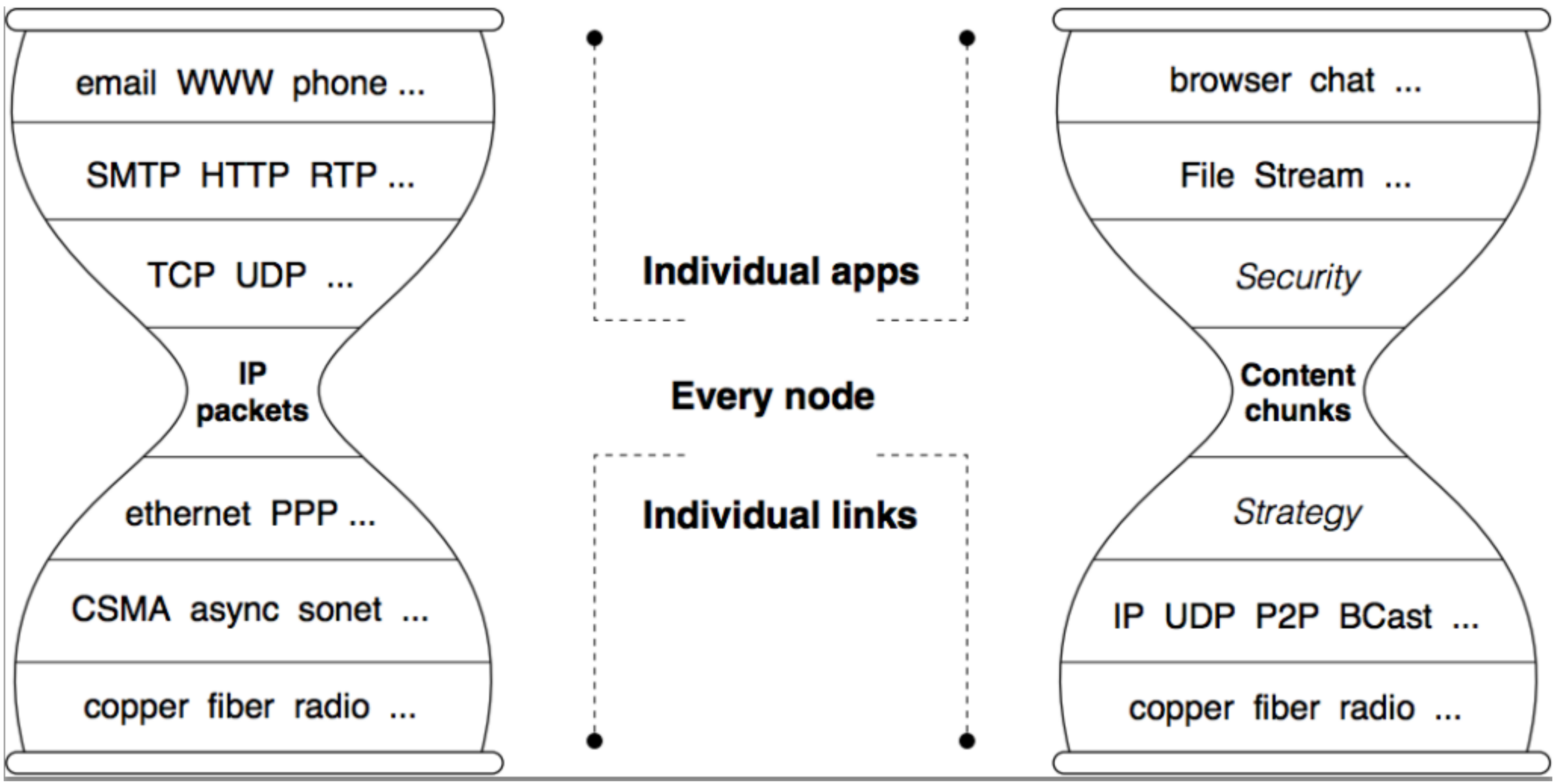
Bowtie Architecture



Hourglass Architecture

The *Protocol Hourglass* idea appears to have originated with Steve Deering. See Deering, S., "Watching the Waist of the Protocol Hourglass", IETF 51, 2001, http://www.iab.org/wp-content/IAB-uploads/2011/03/hourglass-london-ietf.pdf. See also Akhshabi, S. and C. Dovrolis, "The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture", http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p206.pdf.

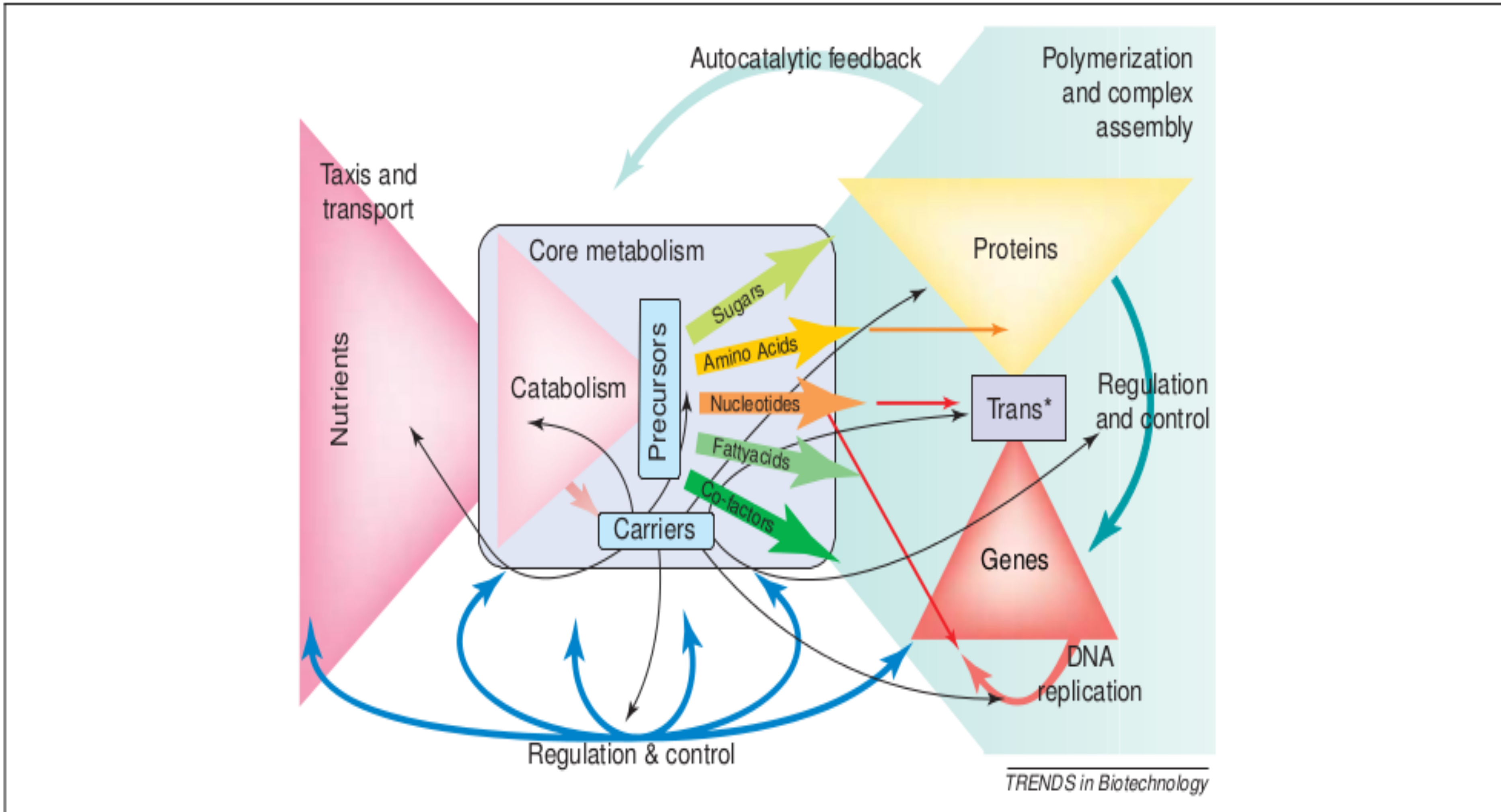
NDN Hourglass



See Named Data Networking, http://named-data.net/

In Practice Things are More Complicated

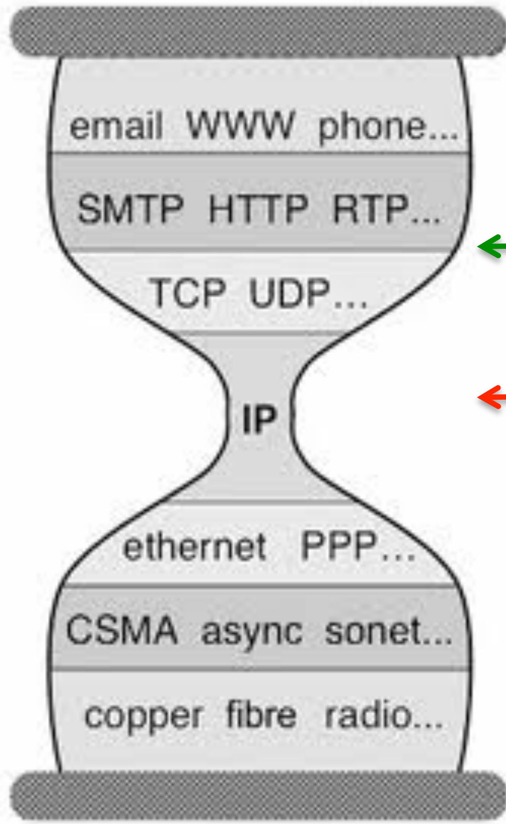
The Nested Bowtie Architecture of Metabolism



See Csete, M. and J. Doyle, "Bow ties, metabolism and disease", TRENDS in Biotechnology, Vol. 22 No. 9, Sept 2004

Hourglasses and SD$^{(a/e)}$?

Open Loop Control + s/w + Moore's Law →
Randomness, Uncertainty, and Volatility



OL/SDN

CP/SDN

OF/SDN?

- OF/SDN?
- CP/SDN makes existing control planes programmable
- OL/SDN is an application *from the perspective of the Internet's waist*

Biology versus the Internet

Similarities

- Evolvable architecture
- Robust yet fragile
- Layering, modularity
- Hourglass with bowties
- Dynamics
- Feedback
- Distributed/decentralized
- ***Not*** scale-free, edge-of-chaos, self-organized criticality, etc

Differences

- Metabolism
- Materials and energy
- **Autocatalytic feedback**
- Feedback complexity
- Development and regeneration
- >3B years of evolution

An Autocatalytic Reaction is one in which at least one of the reactants is also a product. The rate equations for autocatalytic reactions are fundamentally non-linear.

Summary

- Simplicity Rules
- Cloud, NFV, and SD$^{(a/e)}$ are key pieces of the same puzzle
  - Noting telcos have an additional data plane "workload" scaling problem
  - Component Reuse
- However....These systems are rapidly growing in complexity
  - "Software" is a major complexity driver; this effect is accelerating
  - Emergent constraints dominate
    - "Interactions" – Cloud, NFV, and SD$^{(a/e)}$
- Deep understanding of the complexity built into these systems, while quite a ways off, will be critical to our ability to design and operate the CSNSE "networks" of the very near future
  - Cyber-physical on the immediate-term horizon
    - Automation such as we're seeing in cloud is a precursor
- Quite a bit of the ground work needed to understand these systems is being worked out now in a deeply multi-disciplinary way
  - Systems Biology, Control and Information Theory, …
  - Lots of work to do (early days)
  - Engineers are always first to the party: Come join the exploration!

Q&A

Thanks!