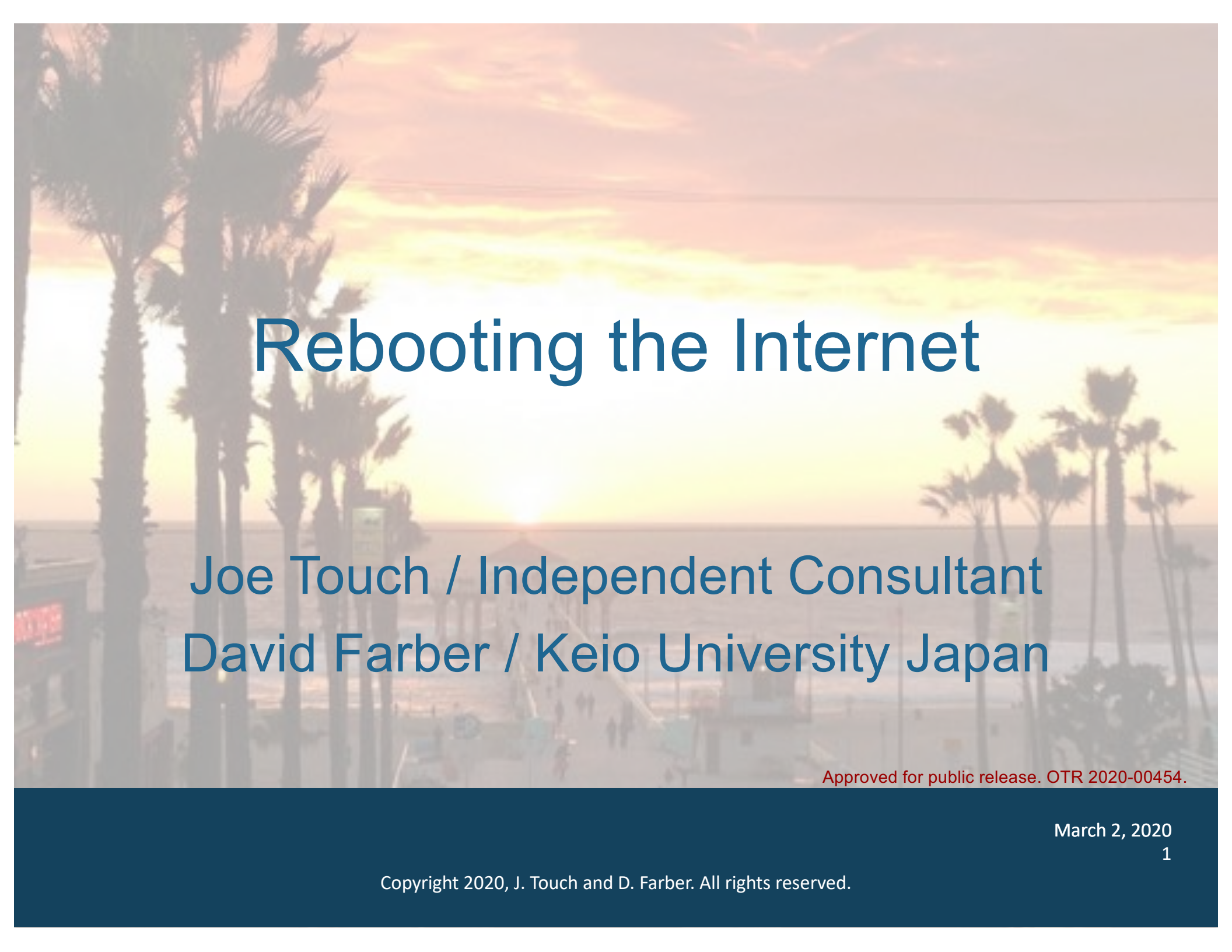




Rebooting the Internet

Joe Touch / Independent Consultant

David Farber / Keio University Japan

Approved for public release. OTR 2020-00454.

March 2, 2020

1

Overview

- Introduction: The Need to Reboot
- Internet Pros and Cons
- What if...
- Networking 2.0 – or 101
- The Challenge of Transition
- Conclusion

Introduction

March 2, 2020

3

Copyright 2020, J. Touch and D. Farber. All rights reserved.

Build One to Throw Away

- “The management question, therefore, is not *whether* to build a pilot system and throw it away. You *will* do that. [...] Hence *plan to throw one away; you will, anyhow.*”
 - Fred P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, 1975.



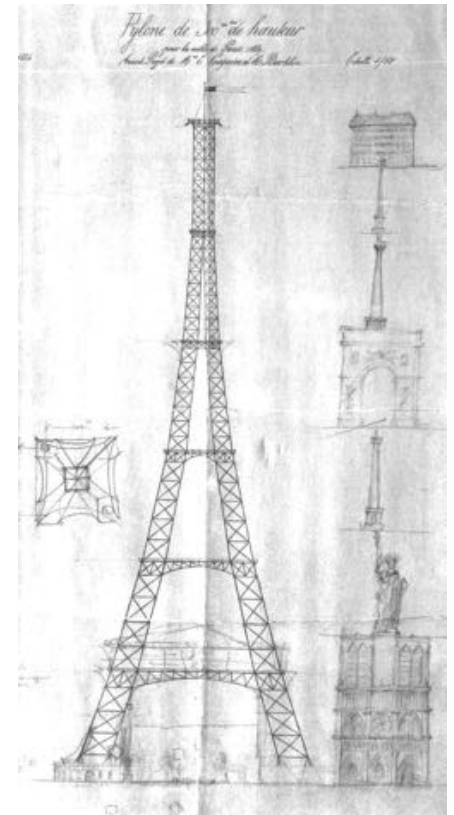
Examples

- Revisions
 - Incremental, in-situ, update/extend
- Reboots
 - Replacement
- “Clean slate” approaches
 - But start with an idea,
not just an eraser



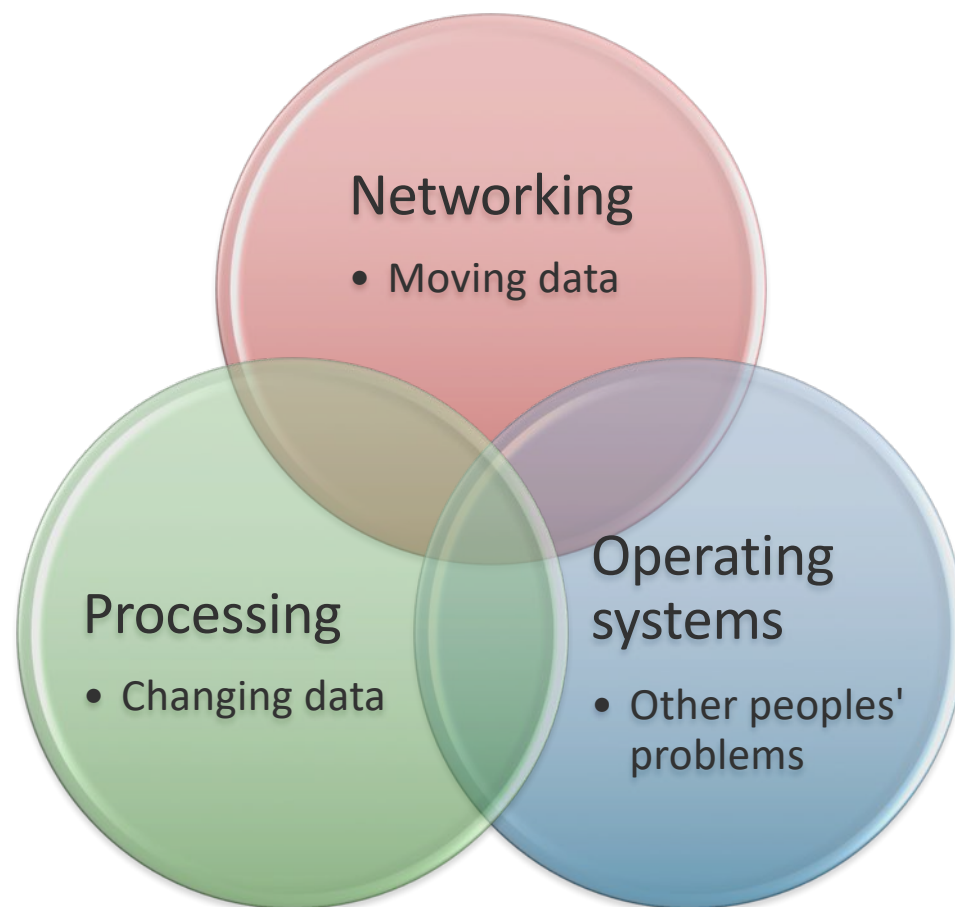
What is Architecture?

- A description of a system through:
 - Its components
 - Their behavior and properties
 - How they interact with each other
 - How they behave as a whole



Intertwined Components

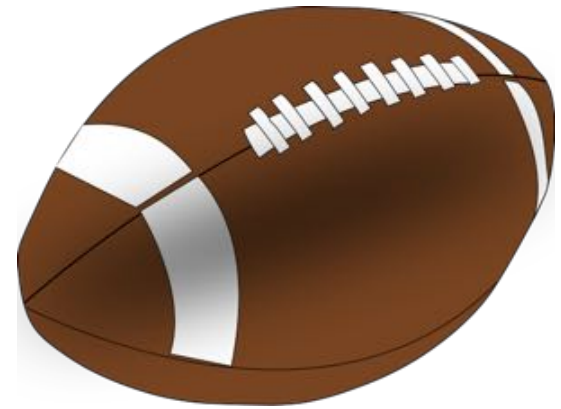
- Co-design is an opportunity...
 - HW/SW
 - Inboard/outboard
 - Library/app
- ...that can be painful to ignore
 - Hard to retrofit
 - Hard to correct



The Need to Reboot is Fundamental

- You can't always get what you want...
 - That was long ago...
- Sometimes, you get what you need
 - And your needs have changed

Lucy's Principle



- Goals change
 - Or the football is moved
- Internet history
 - Interpersonal comm, telework
 - NOT nuclear survivability (Baran's was, though)
 - Not voice, video, etc., not scale per se
- Telecom
 - Text message (telegraph) to voice (phone) to email to web to video

Outdated questions

- Compress video over phone
 - Voice phone
 - Sign language
- Compressed voice
 - Under 8KHz
 - Under 7 bits/sample



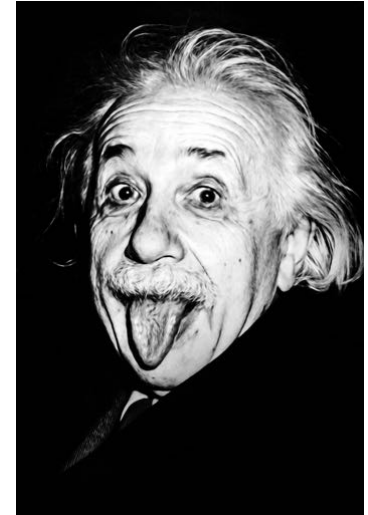
Farber's Principle



- Resources change
 - Memory / compute = microcode, compressed RAM
 - BW / compute = compressed data
 - BW / disk = disk caching
 - SIMD special vs. GPU
 - Prevalent mem/CPU errors vs. coding errors mostly

Einstein's Principle

- The answers change
 - Even when the questions don't
- Architecture changes
 - Understanding of the components
 - How they relate



Hindsight is 20/20

- Design to what you know
 - Rarely know what you're creating before/as you create it
- Now you know better
 - Better understanding of parts, whole, and relationships



Brooks had second thoughts...

- Before OS/360 was Stretch
 - Multiprogramming, memory protection, interrupts, 8-bit byte, pipelining, memory interleaving
 - Bit off a LOT
- V2: Brooks
 - "This I now perceived to be wrong, not because it is too radical, but because it is too simplistic. The biggest mistake in the 'Build one to throw away' concept is that it implicitly assumes the classical sequential or waterfall model of software construction."
-- The Mythical Man-Month, 20th Anniversary Edition, pg. 265
- And his “second system effect”:
 - An architect's first work is apt to be spare and clean. He knows he doesn't know what he's doing, so he does it carefully and with great restraint.
 - As he designs the first work, frill after frill and embellishment after embellishment occur to him. These get stored away to be used “next time.”
 - Sooner or later the first system is finished, and the architect, with firm confidence and a demonstrated mastery of that class of systems, is ready to build a second system. This second is the most dangerous system a man ever designs.
 - The general tendency is to over-design the second system, using all the ideas and frills that were cautiously sidetracked on the first one. The result, as Ovid says, is a "big pile.”

A Review of Reboots

- *Green Day's American Idiot*
 - 2005 masters for *Cigarettes and Valentines* stolen
- Tacoma Narrows bridge
 - (Galloping Gertie)
- Shinkansen
 - vs. Acela
- OSs
 - DOS/Windows, Apple 68K/x87
- The Internet itself
 - NCP to TCP/IP (disruptive)
 - Classful to CIDR (non-disruptive)
 - IPv4 to IPv6 (partly disruptive)



Internet Pros and Cons

March 2, 2020
16

The Internet is...

- Packets
 - Variable length messages
 - Self-addressed (source and dest)
- Global addresses
 - With some structure
- Local forwarding decisions
 - Allowing incremental deployment



Successes

- One ring to rule them all
 - Single unifying layer
- Best effort is good enough
 - Ubiquity over performance
- Local decisions for global behavior
 - Allows local tuning, local config/deployment



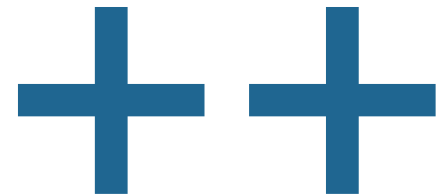
Opportunities to Improve

- Little things

- Email: FROM return address vs. ORIGIN source
- TCP closer holds state, not opener
- Fixed vs. extensible fields (TLVs)
- Address allocation (big chunks too early)
- Endian choice

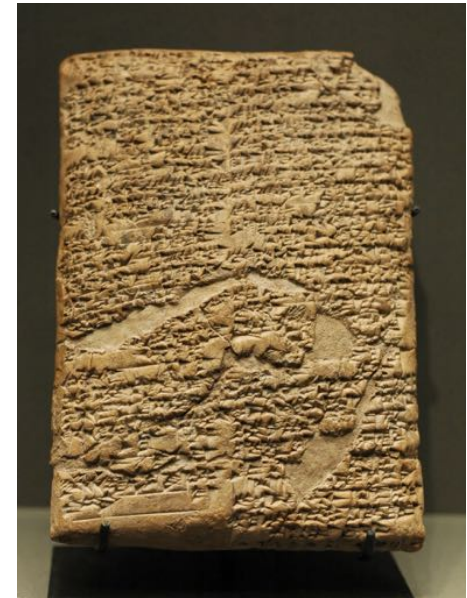
- Big things

- 7-layer model
- Local subnetting on protocol design
- Security boundaries (vs. Multics)
- Security as binary (all or nothing)



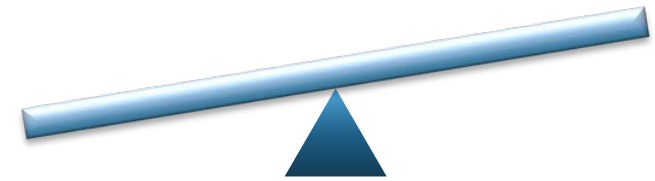
Good Architecture

- Like a good list
 - Nothing left to add or remove
 - Similar size/scope of components
 - No component overlap
- Like a good model
 - Provides analogy: conceptual homomorphism
 - Describe existing behavior (predictably)
 - Predict new behavior



Good Design = Balance

- Intel/MS TPM
 - Multicompany security effort
- Multics
 - Integrated HW/SW design
 - Ambition = late, replaced by Unix (“the enemy of good enough”)
- Microcode
 - HW/SW trade, complexity trade
 - Programmable but nearly impossible to debug



WHAT IF...

March 2, 2020
22

Copyright 2020, J. Touch and D. Farber. All rights reserved.

Why “what if”?

- Challenge assumptions
 - Because they become constraints
- Challenge context
 - Because they limit what can be imagined
- Challenge expectations
 - Because they make it easy to be content

Consider our assumptions

- Most stub nets are broadcast
 - Ethernet, in particular
- Most sessions are streams of packets
 - Enough to stabilize over dozens of round-trips
- Packets are relatively small
 - Enough to allow sharing on low-BW links (ATM)

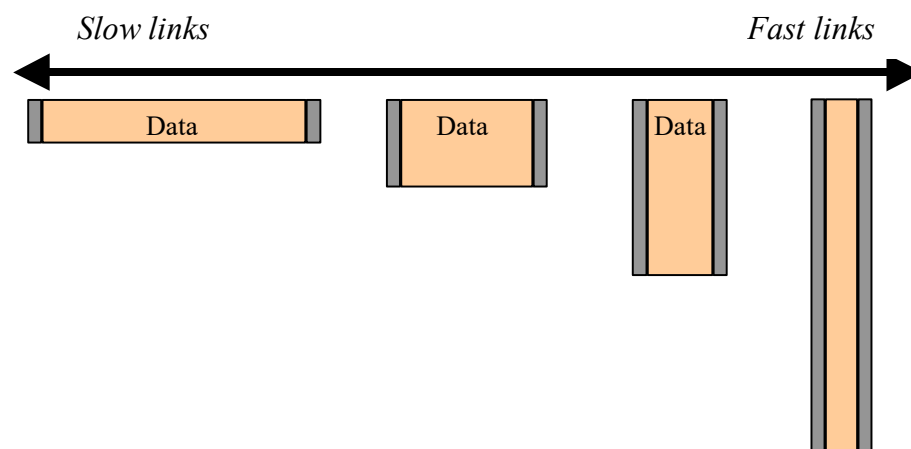
WHAT IF: there were no subnets?

- Currently: rely on broadcast/multicast
 - L2 transit discovery
 - Device configuration
 - Resource discovery
 - Efficient multipoint data distribution
- Suppose: links were only unicast
 - Would we need LANE (LAN emulation) everywhere?

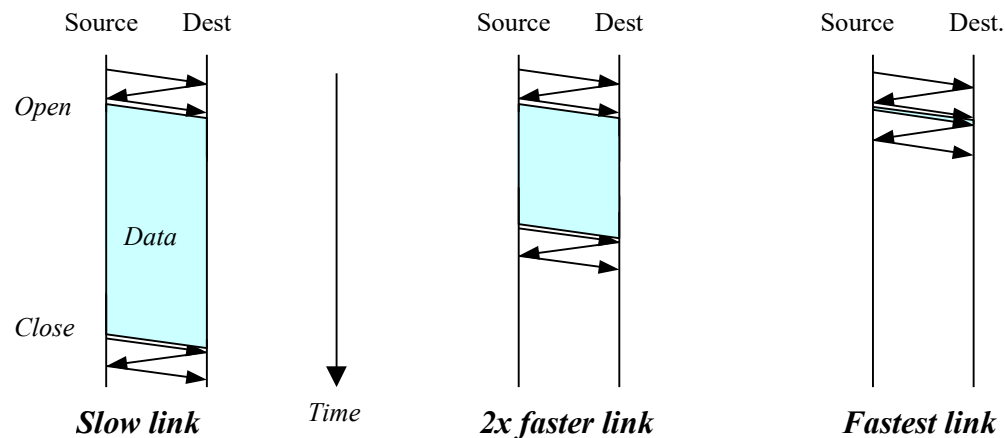


BW foreshortens messages

- Packets spend less time “on the wire”



- Still affected by transit



WHAT IF: flows were one packet?

- Currently: long-term dominates
 - Most flows are short, but...
 - Most packets are part of long flows
 - “Steady-state” dominates
- Suppose: packets were solo
 - Most flows were one packet
 - Would packets be bigger?...



WHAT IF: we used collossograms?

- Currently: MTU hasn't scaled
 - 80% ≤ 512 in 1969-2000
 - 80% $\leq 1,500$ in 2000-2005
 - 80% $\leq 1,200$ now (tunnel effects)
- Suppose: packets were huge
 - Constant time (M. Mathis' request)
 - Just large ($> \text{RTT} \cdot \text{BW}$)
 - Would “store and forward” work?



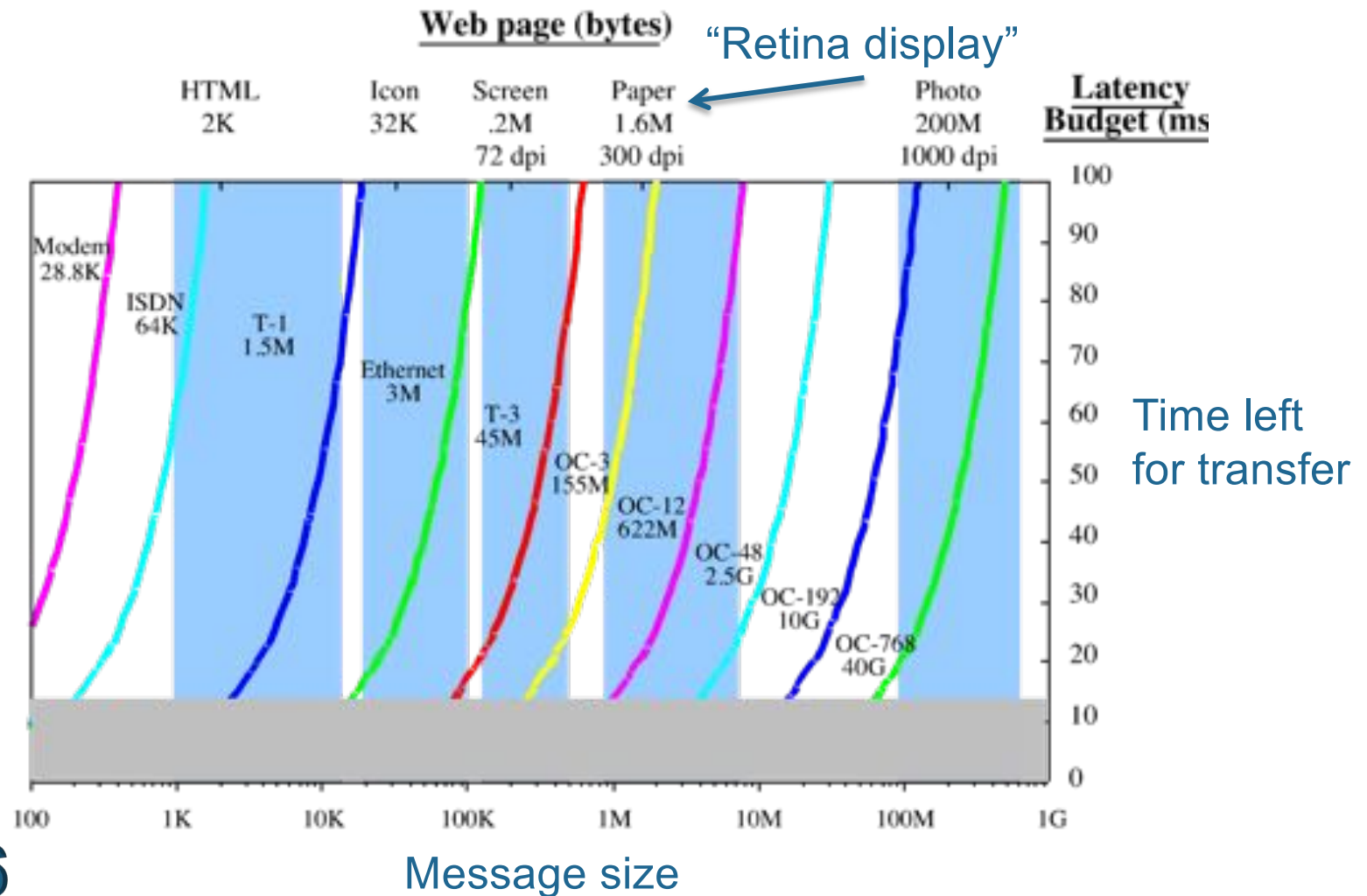
Consider our context

- Limited CPU (until 1990s)
 - IMPs as outboard processors
- Limited memory / disk (until 2000s)
 - Compression, on-demand access
- Limited BW (until 2010s)
 - Compression, synchronous exchanges
- Limited addresses (until IPv6?)
 - NATs
- Error-prone (?)
 - Error detection and correction throughout

WHAT IF: BW were a resource?

- Currently: BW limits usage
 - Recently less a last-mile issue
 - Still server limited
("speedtest" vs. real content)
- Suppose: BW were abundant
 - Already prefetch / can pre-push
 - What performance can BW buy?

What BW do we need?

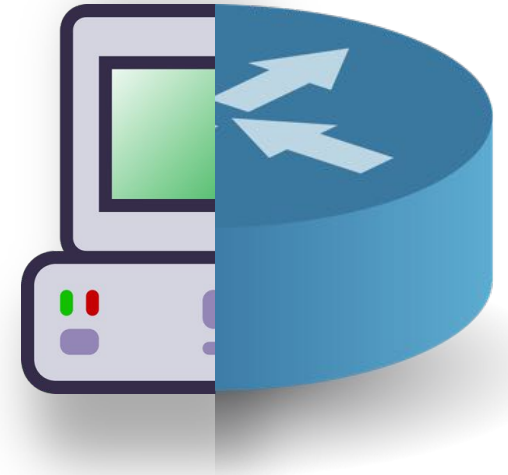


Circa 1996

March 2, 2020
31

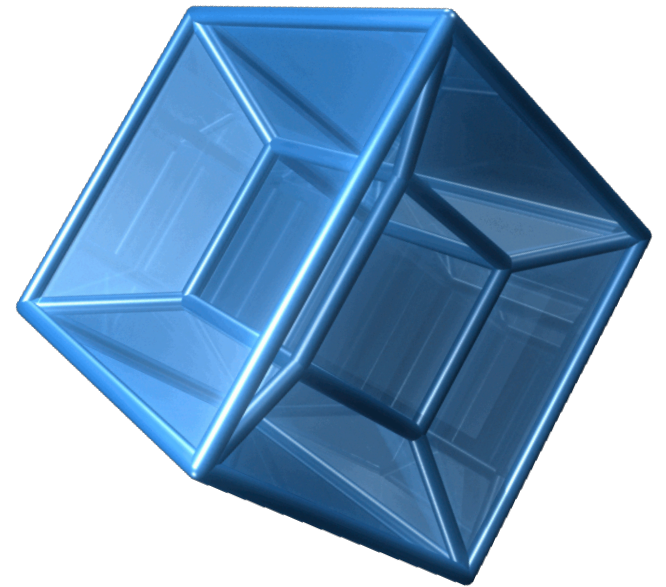
WHAT IF: hosts lacked addresses?

- Currently: $\text{addr:host ratio} \geq 1$
 - Multihoming
 - OS/server parallelism
- Suppose: $\text{addr:host ratio} < 1$
 - Do all hosts need an address?
 - Is NAT a special case or more general?
 - Can ports or other identifiers suffice for 'intra-host' demultiplexing?



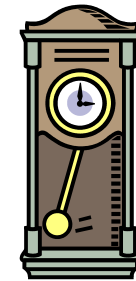
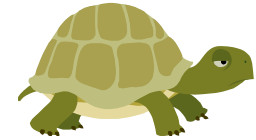
WHAT IF: names had dimensions?

- Currently: static, global
 - Coarse structure
 - Topology (IP), semantics (IP, Ethernet)
- Suppose: other properties
 - Time (duration of validity)
 - VPN-IDs (context)
 - Security level (like original Internet)

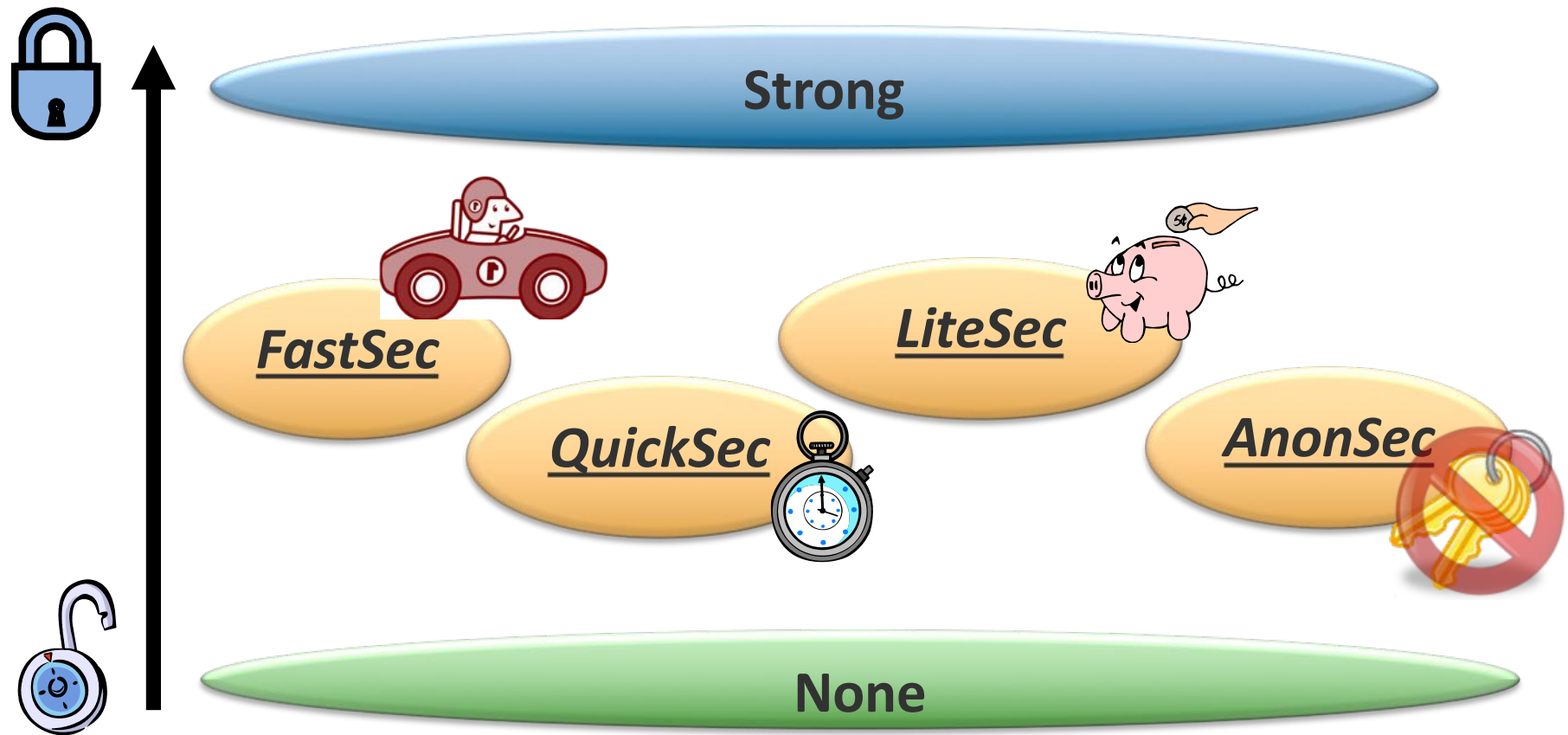


Security is expensive...

- Currently: high bandwidth cost
 - Factors of 2-4 slowdown
- Currently: high latency
 - Store-and-forward delays, block delays
- Currently: high processing load
 - CPU load, additional hardware, battery life
- Currently: requires key infrastructure
 - The impediment to deployment



WHAT IF: security were cheaper?



Cases for Weak Security

- Suppose: anonymous
 - No ID, e.g., IETF BTNS
 - Security without PKI *but weaker*
- Suppose: lite
 - Lightweight, using layered checks
 - Low CPU/energy impact (esp. under DOS attack) *but weaker*
- Suppose: fast
 - High speed, using parallel chaining
 - Low performance impact *but weaker*
- Suppose: quick
 - Low-latency, over shorter blocks than a packet
 - Low delay with higher overhead or *weaker with same overhead*

The Cautionary Tale of Maslow's Hammer Emporium

- What if != why not
 - E.g., Clean Slate, Active Nets, SDN, ICN/CCN, DTN, etc.
 - Contrarianism != answer
 - COULD != SHOULD



got nails?

Other benefits of WHAT IFs:

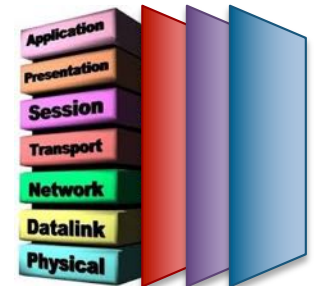
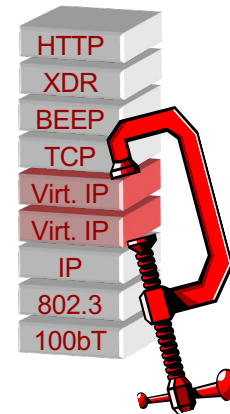
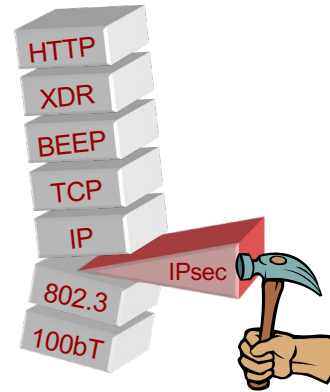
- Buried in these and other questions:
 - An interesting project
 - An interesting paper
 - A thesis
 - A sabbatical

Networking 2.0 – or 101

An example of “WHAT IF”

Current architecture fails

- Shim layers
 - MPLS, IPsec, HIP
 - ARP, RARP, IS-IS
- Recursive layers
 - IP-in-IP, IPsec tunnels
 - IP over UDP, Ethernet over IP
- Missing “planes”
 - Control, management, naming



It's about the architecture

- The 7-layer model never was
 - And “the cake is a lie”
- Protocols are ad-hoc
 - “on the wire” vs. state sharing
- Design as art – at best (perhaps “collage”)
 - Where's the science?
 - What is the foundation?
 - What are the principles?

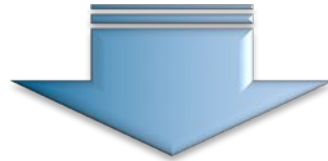
Where are our first principles?

A First-Principles Approach to Computer Networking

- Start with information theory
 - 2-party shared state
 - Extend to multiparty, multihop, multi-scale, ...
- Leverage the key tools of CS
 - Abstraction
 - Recursion
- Result is an integrated model
 - Layering, forwarding, name resolution
 - Agnostic to details (or “allows many instances”)

Where are our “3 laws of motion”?

- (Shannon) Communication
 - Shared state between two (given) parties that are directly connected by a perfect channel.



- Networking
 - Relax each of the underlined phrases above
 - Support heterogeneity and scale

Logical progression

Communication

Multiparty

Layering

Naming

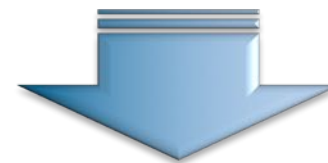
Recursion

Communication to Protocols

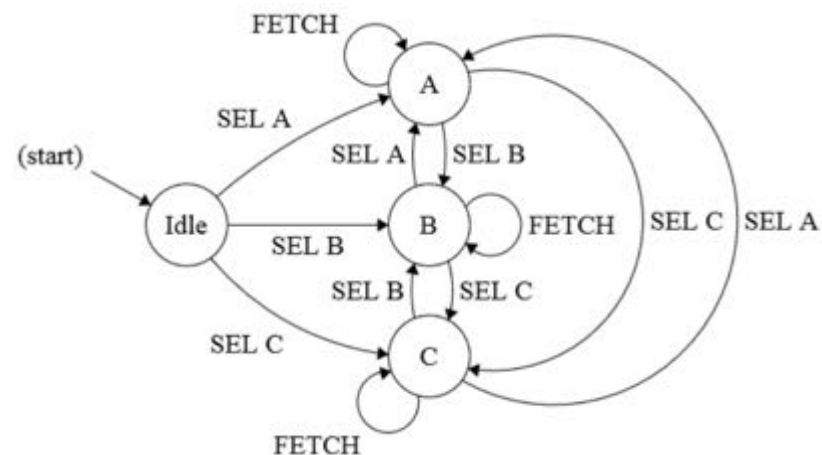
- Perfect channel
 - Start with Shannon 2-party, no errors



- Imperfect channel
 - Add errors

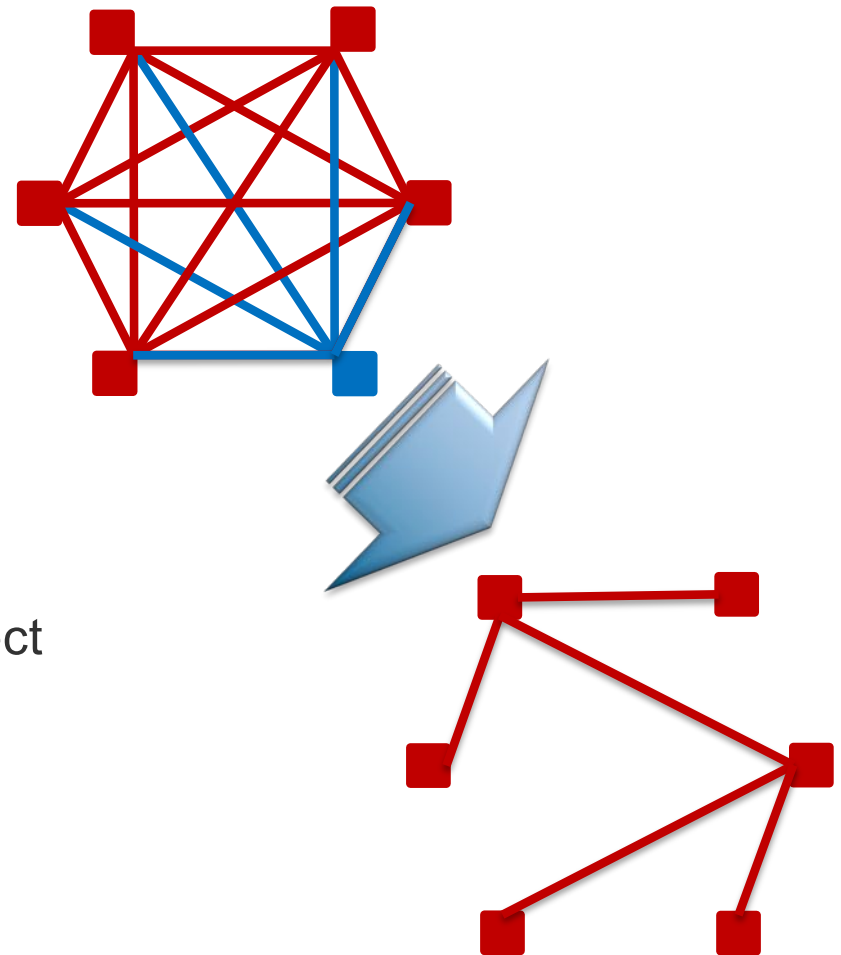


- Making the channel real
 - From theory to practice
- Finite state machines
 - Limited Turing Machine



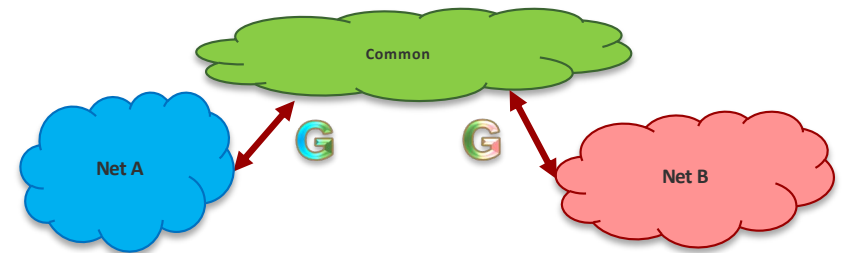
Multiparty to Relaying

- Multiparty means names
 - TO: broadcast; FROM: shared
- Explicit shared channels
 - Fermions and bosons
 - Polling, schedules
 - TDMA, FDMA, CDMA
- Implicit shared channels
 - Token passing, collision sense/detect
 - Token ring, Ethernet
- Relaying
 - Emulated sharing

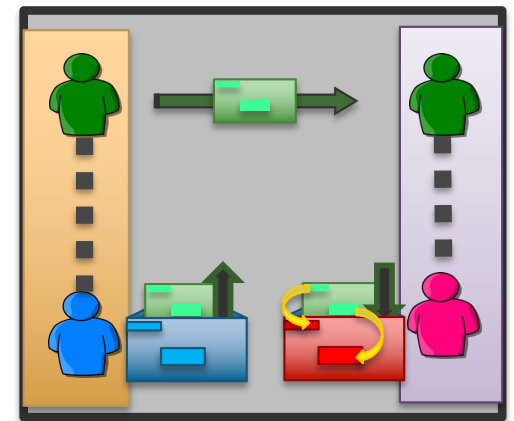


Layering

- How we support heterogeneity, scale
 - Via translation (convert)
 - Via encapsulation (stack)



- Combine the two
 - Encapsulate via translation
 - Payload encapsulation with header translation



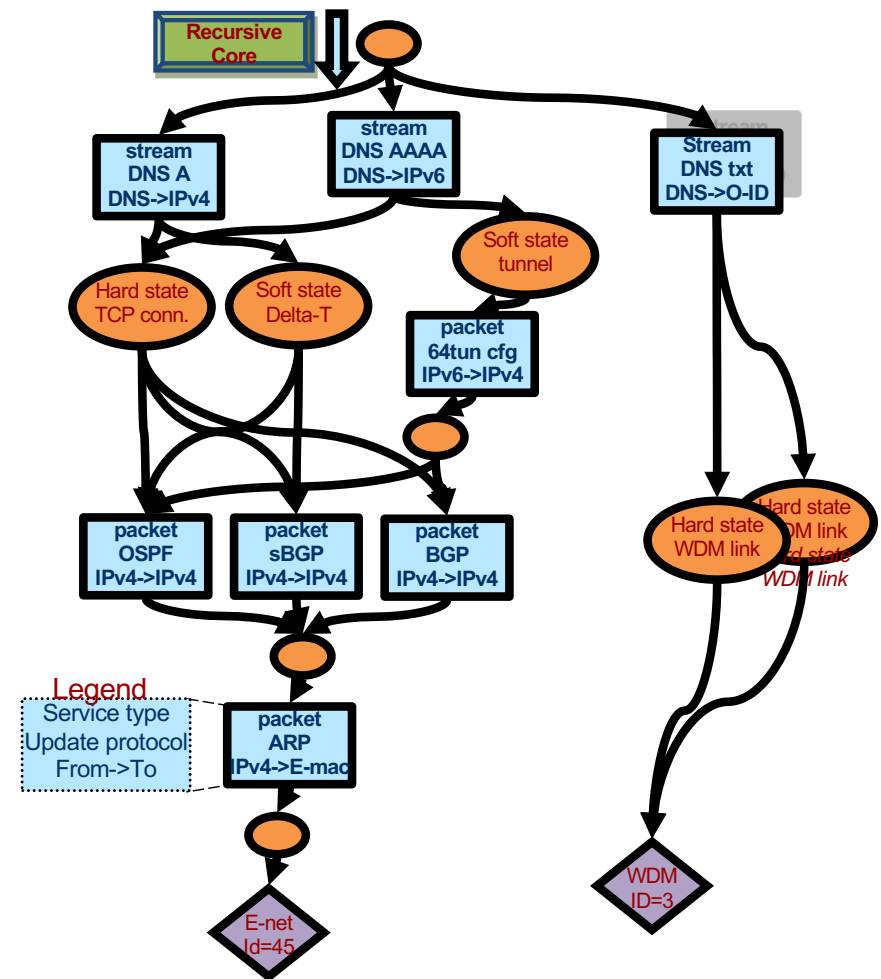
Naming

- Purpose and structure of names
 - Translation, assignment, switching
- Resolution mechanisms
 - Fixed, local, flat map, hierarchical map
- Inside and outside names
 - Sockets, port numbers, role of 'bind'
 - Strong vs. weak protocol names
- Automatic naming
 - Self naming, duplicate detection



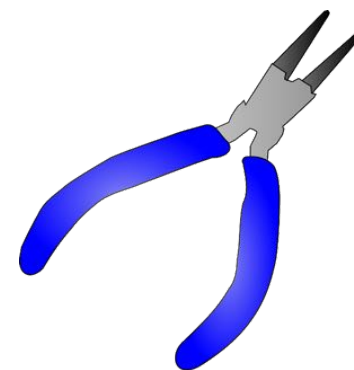
Recursion

- Layering as recursion
 - DAG represents possible paths
- Walking the DAG
 - FSMs and translation tables
 - Stack isn't software
 - It's how headers record the DAG path
- Routing protocols
 - Populating the DAG
- Forwarding as tail recursion
 - Forwarding as tunneling
 - Routing as a promise



The rest is optimization

- Dealing with deficiencies
 - Integrity (corruption, loss, **tampering**)
 - **Authentication**
 - **Privacy**
- Performance
 - Time, space, energy
- Emulation
 - Circuit/wire, transactions, translation



First-Principles Architecture

- Describes current architectures as degenerate cases
 - OSI
 - Internet
- Extends to fix deficiencies
 - Same mechanism at all layers
 - Integrates layering, forwarding, naming
- Predicts new capabilities
 - “Late-binding” layering that adapts
- Explains why
 - Rather than just describing how

The Challenge of Transition



March 2, 2020
52

Evolve vs. Replace

- Evolve
 - HTTP 1.2, 2.0
 - MPTCP, QUIC
 - Emulation libraries / firmware
 - Intel: x86 to x86-64
- Replace
 - MULTICS changed ASCII versions on “Flag day” (June 14, 1966)
 - NCP to TCP/IP (Jan 1, 1983)
 - Apple: 68K to PowerPC to x86 to ARM
- Or combine the two?
 - Run over Internet over IP?
 - Or over UDP?
 - Or over HTTP?

Competing Interests

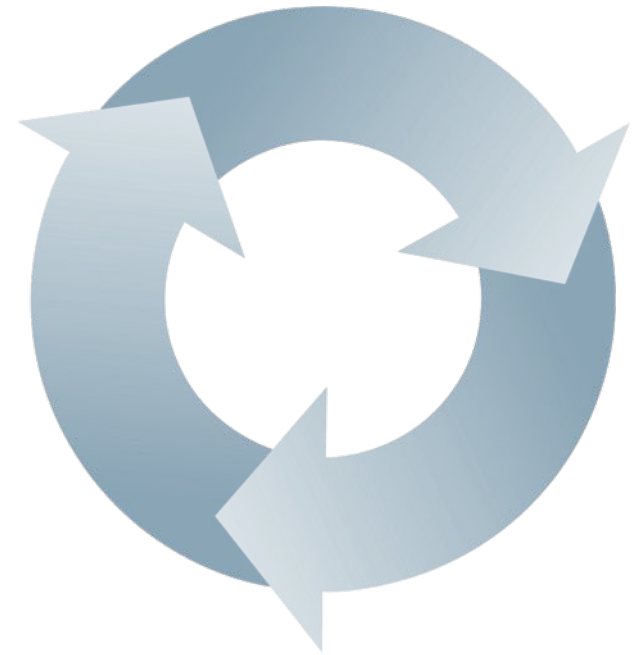
- Differentiation vs. interoperability
 - Design to break into a market or corner it
- Now vs. future
 - Design for just now or into tomorrow
- The Borg vs. the Hammurabi (Linux vs. BSD)
 - Design for compatibility or stability
- Good enough vs. fast and furious
 - Design for correctness or speed

People and Money

- Innovator's dilemma
 - Large = entrenched, unlikely to abandon market
 - Small = willing to take risks, less downside
- Funders dilemma
 - NSF – one student at a time
 - DARPA – results, not research
 - Industry – products, not research
- Alternate facts
 - Relativism as a disease

Cycles

- Pendulums of solutions
 - Integrated vs. outboard I/O
 - Support for errors vs. assume none
- Ambiguous answers
 - Extensible vs. fixed fields
 - CISC vs. RISC



Breaking The Cycle

- The ice man cometh no more
 - Some changes abandon you
- From pistons to jets
 - Some changes are thrust upon you

Conclusions

Sisyphus was right

- There's no such thing as 'done'
- The second time isn't magic either
 - “When he does his third and later ones, his prior experiences will confirm each other as to the general characteristics of such systems, and their differences will identify those parts of his experience that are particular and not generalizable.” – Brooks



Entropy Rules

- Easy to diverge from original plan
 - Second++ system effect
- Too many cooks...
 - Different goals, viewpoints
- Dust accumulates
 - General bit-rot

Will Anyone Listen?

- Not the ISOC
 - IETF patches
 - IAB doesn't architect
- NSF
 - Only one student at a time
- DARPA
 - “Revolution not evolution”
 - Celebrates anniversaries



Acknowledgements

- Farber – ccrc.keio.ac.jp
 - D. Farber, “1960s and Future”, WDF 12/6/2019 50th Aniv Internet, Shenzhen
 - D. Farber, Keio Univ , SFC Faculty talk, 11/18/2019
- Touch - www.strayalpha.com
 - J. Touch, “[A First-Principles Approach to Computer Networking](#),” Tutorial at IEEE CCNC 2017, IEEE CCNC 2018.
 - J. Touch, J. Day, “[‘One Packet Only’ and Other Curiosities: Alternate Axioms that Shake Up the Hourglass](#),” with J. Day at IEEE Workshop on Computer Communications (CCW), Steamboat Springs, CO, Oct. 2008.
 - J. Touch, “[One Ring to Rule Them All: Unifying Computation and Communication](#),” Panel at IEEE Rebooting Computing Workshop, Oct. 2016.
 - J. Touch, “[The Case for Weaker Network Security](#),” keynote, IEEE Computer Communications Workshop (CCW), Niagara Falls, Canada, Nov. 2013.
 - J. Touch, “[The Optical Turing Machine](#)”, IEEE Rebooting Computing Workshop 2016.
 - J. Touch, “[Recursion and the Transport Tussle](#)”, PfLDNet 2010, Lancaster, PA, Nov. 2010.
 - J. Touch, “[Understanding Latency - a Root Cost and Mitigation Approach](#),” Tutorial at IEEE Globecom 2014, IEEE CCNC 2016, ACM Sigcomm 2017.