How in the Heck Do You Lose a Layer!?

Future Network Architectures Workshop

University of Kaiserslautern, Germany

John Day

2012

OSI only had a Network Layer, but
the Internet has an Internet Layer!

- Noel Chiappa, 1999

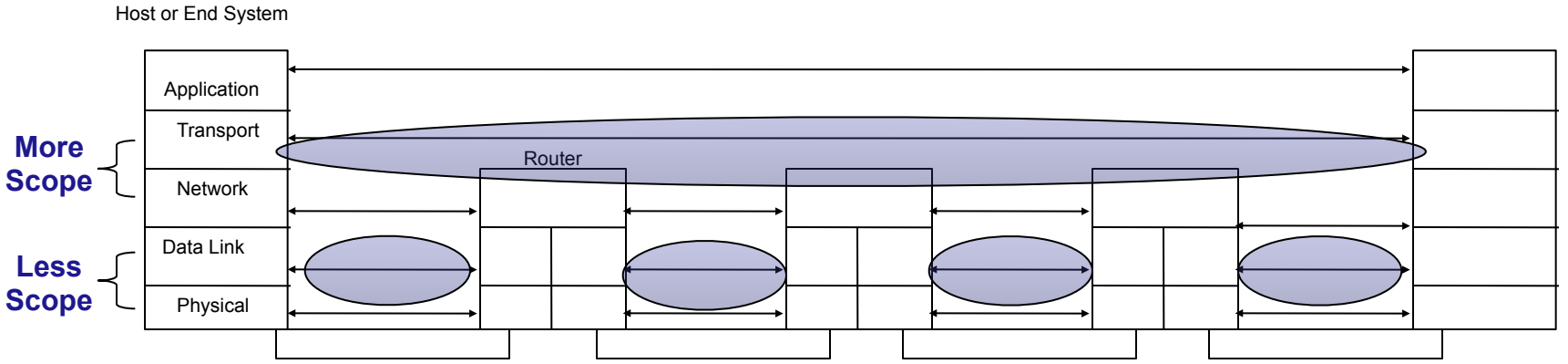
Preamble

- A Couple of Remarks on the Nature of Layering
- The advent of packet switching required much more complex software than heretofore, and so the concept of layers was brought in from operating systems.
- In operating systems, layers are a convenience, one design choice.
- In networks, they are a necessity.

The (really) Important Thing about Layers

(From first lecture of my intro to networks course)

- Networks have loci of distributed shared state with different scopes
- At the very least, differences of scope require different layers.
- It is *this* property that makes the earlier telephony or datacomm "beads-on-a-string" model *untenable*.
  - Or any other proposal that does not accommodate scope.
- If there are multiple layers of the same scope, their functionality must be independent.
  - This has always been understood.



1972 Was an Interesting Year

- Tinker AFB joined the ‘Net exposing the multihoming problem.
- The ARPANET had its coming out at ICCC ‘72.

- As fallout from ICCC 72, the research networks decided it would be good to form an International Network Working Group.
  - ARPANET, NPL, CYCLADES, and other researchers
  - Chartered as IFIP WG6.1 very soon after

- Major project was an International Transport Protocol.
  - Also a virtual terminal protocol
  - And work on formal description techniques
- But a major 3-sided war was just breaking out

War on All Sides

- The Phone Companies don't like the new model because an end-to-end transport relegates them to a commodity business, thus not giving them exclusive claim to "value-added services" or "services *in the network*."

- IBM (with 80% of the computer market) doesn't like the new model because it has a hierarchical network architecture, SNA.

- The other computer companies (especially the minicomputer) love the new model because it plays to their strength and because it nails down the other two!

- This War (and it was) dominates everything for the next 2 decades and really is still going on.

Meanwhile Back at INWG

There Were Two Proposals

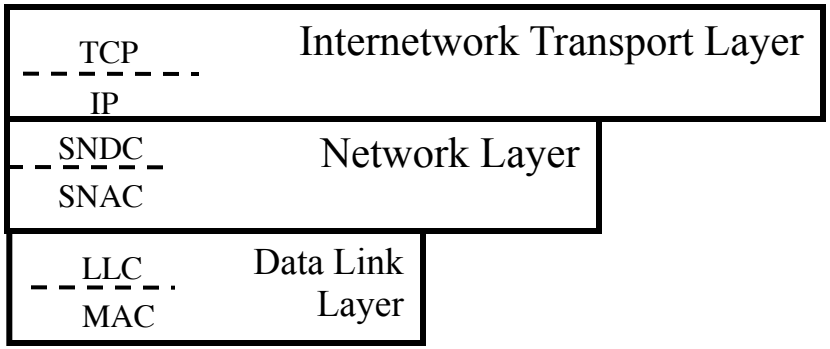
- INWG 37 based on the early TCP and
- INWG 61 based on CYCLADES TS.
- And a healthy debate, see Alex McKenzie, "INWG and the Conception of the Internet: An Eyewitness Account" IEEE Annals of the History of Computing, 2011.
- Two sticking points
  - How fragmentation should work
  - Whether the data flow was an undifferentiated stream or maintained the integrity of the units sent (letter).
- These were not major differences.

After a Hot Debate

- A Synthesis was proposed: INWG 96
- There was a vote in 1976, which approved INWG 96.
- As Alex says, NPL and CYCLADES immediately said they would convert to INWG 96; EIN said it would deploy only INWG 96.
- "But we were all shocked and amazed when Bob Kahn announced that DARPA researchers were too close to completing implementation of the updated INWG 39 protocol to incur the expense of switching to another design. As events proved, Kahn was wrong (or had other motives); the final TCP/IP specification was written in 1980 after at least four revisions."
  - Neither was right. The real breakthrough came two years later.
- But the differences weren't the most interesting thing about this work.

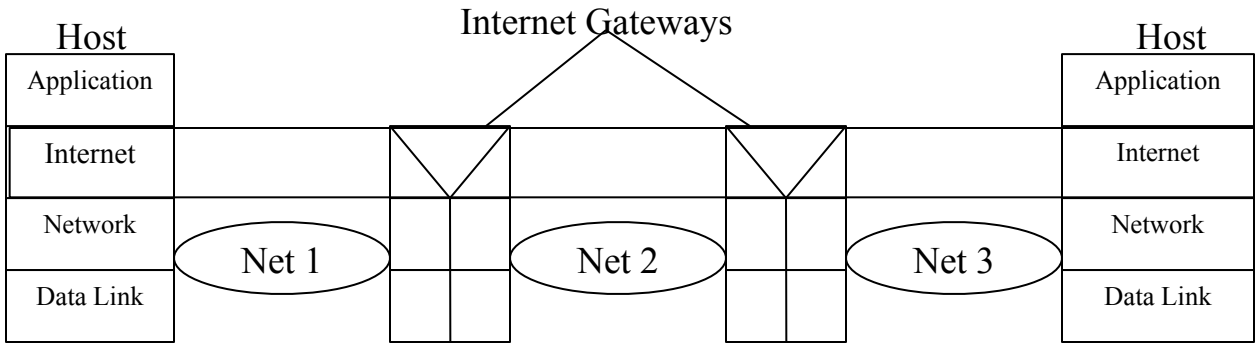
The *Similarity* Among all 3 Is Much More Interesting

- This is before IP was separated from TCP.
- All 3 *Transport* Protocols carried addresses.
- This means that the Architecture that INWG was working to was:



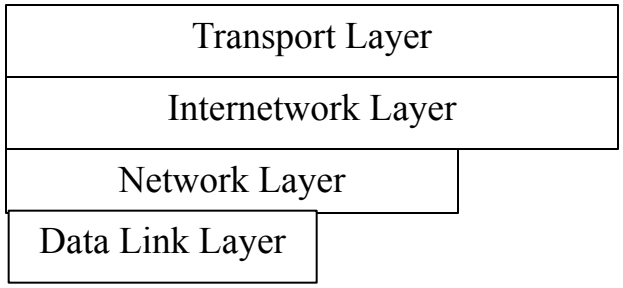
- Three Layers of Different Scope each with Addresses.
- If this does not hit you like a ton of bricks, you haven't been paying attention.
- This is *NOT* the architecture we have.

INWG's Internet Model

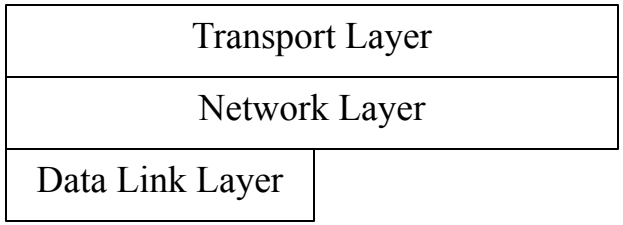


- An Internet Layer addressed Hosts and Internet Gateways.
- Several Network Layers of different scope, possibly different technology, addressing hosts on that network and that network's routers and its gateways.
  - Inter-domain routing at the Internet Layer; Intra-Domain routing at the Network Layer.
- Data Link Layer smallest scope with addresses for the devices (hosts or routers) on segment it connects

Separating IP from TCP



- Shouldn't have changed things much. But it did.
- They thought they needed a flag day to transition from NCP to TCP/IP and when they were done, it looked like this:

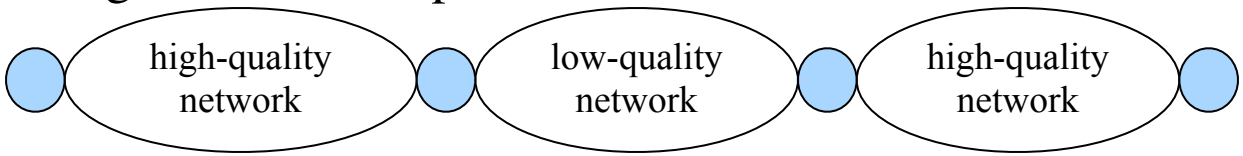


How Did They Lose A Layer?

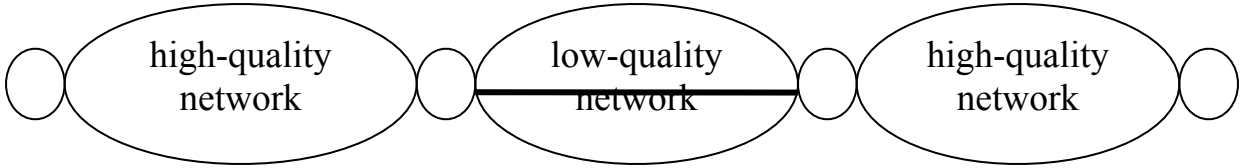
- What Appears to Have Happened:
  - By 1980 or so: A Large Central ARPANET surrounded by LANs
    - ARPANET was a black box run by BBN with its own routing and congestion control.
  - Routers start appearing with Ethernet on one side and ARPANET on the other
    - Called Internet Gateways at the time.
  - Some are directly connected, with Static Routing or Simple Routing schemes
  - As the 'Net expands and the ARPANET shrinks . . .
    - More routers are directly connected but still all under DoD, so not seen as separate networks with distinct routing domains. And not big enough that it is a problem.
  - The Non-ARPANET network is routing on IP as a *Network Layer* protocol!
    - Little overlap between the INWG and the Internet, by now IP is the Network Layer!
  - No one notices that there needs to be a network layer *and* an internetwork layer. And anyway functions don't repeat in layers.
- But Two Layers of Different Scope is Not a Repetition
    - Scope? What is that?
  - The Same Protocol can used to Provide Functions of Different Scope.
    - To refine Dykstra, Functions shouldn't repeat in layers of the same scope

Okay, What Did OSI Do?

- The other major effort at the time.
- The well-known 7-layer model was adopted at the first meeting in March 1978 and frozen. After that, they had to work within that.
- One of the concerns in the Network Layer deliberations was interworking over a less capable network:

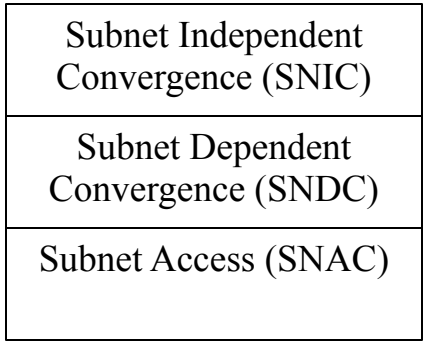


- Would need to enhance the less capable network with an additional protocol

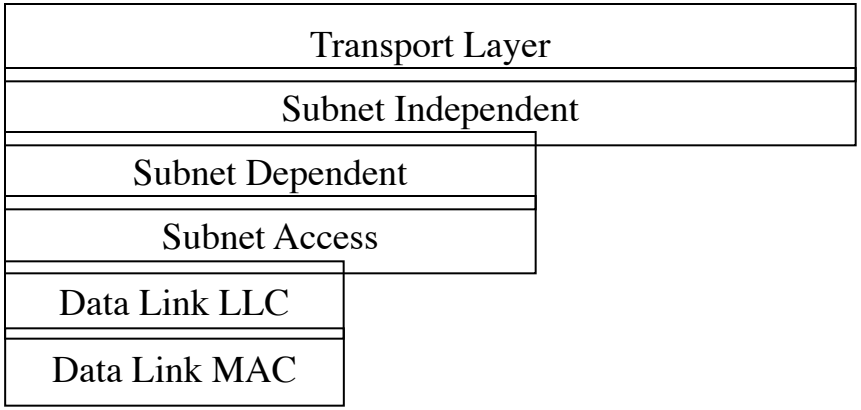


OSI Sub-Divided the Network Layer

- This concern and the recognition that there would be different networks interworking lead OSI to posit three sublayers, which might be optional depending on configuration:



OSI Also Came to the INWG Model



- With a Transport Layer, this is the same as the INWG model.
- OSI stayed the course with a similar split to TCP/IP.
- So OSI had an Internet Architecture and the Internet has a Network Architecture.

- And signs of a repeating structure.

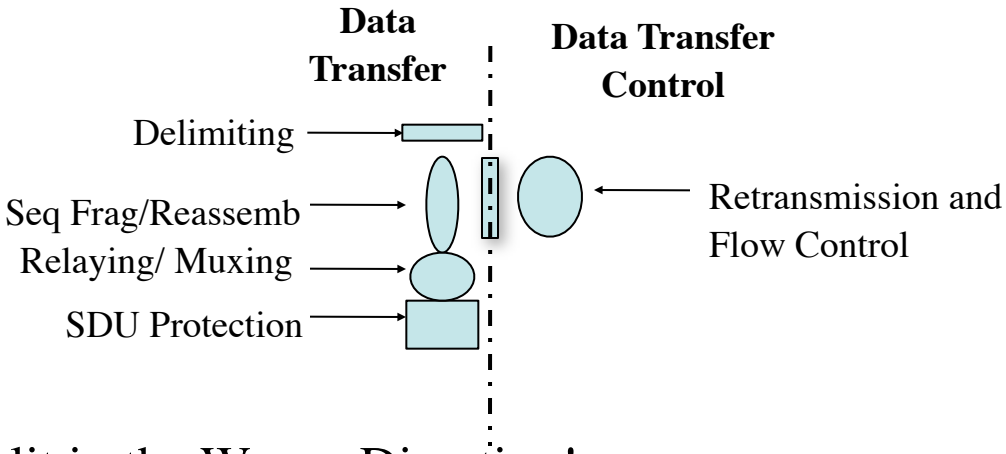
Was Splitting TCP and IP Right?

- The Rules say if there are two layers of the same scope, the functions must be independent for it to work well.
- Error and Flow Control separated from Relaying and Multiplexing are independent. But IP also handles fragmentation across networks.
  - Remember, Don't repeat functions in different layers of the same scope?
- Problem: IP fragmentation doesn't work.
  - IP has to receive all of the fragments of the same packet to reassemble.
  - Retransmissions by TCP are distinct and not recognized by IP.
    - Must be held for MPL (5 secs).
    - There can be considerable buffer space occupied.
- There is a fix: MTU Discovery.
  - The equivalent of "doc, it hurts when I do this!" "Then don't do it."

But it is the *Nature* of the Problem

that is Interesting

- The problem arises because there is a dependency between IP and TCP. The rule is broken.
  - It tries to make it a beads on a string solution.
- A Careful Analysis of this Class of Protocols shows that the Functions naturally cleave along lines of Control and Data.



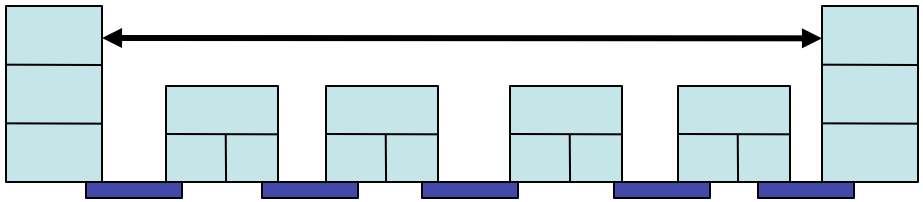
- TCP was split in the Wrong Direction!
  - It is one layer, not two.
  - IP was a bad idea.

A Chance to Get Things on Track

- We knew in 1972, that we needed Application Names and some kind of Directory.
- Downloading the Host file from the NIC was clearly temporary.
  - And eventually we would have many more applications than the basic 3.
- When the time came to automate it, it would be a good time to introduce Application Names!

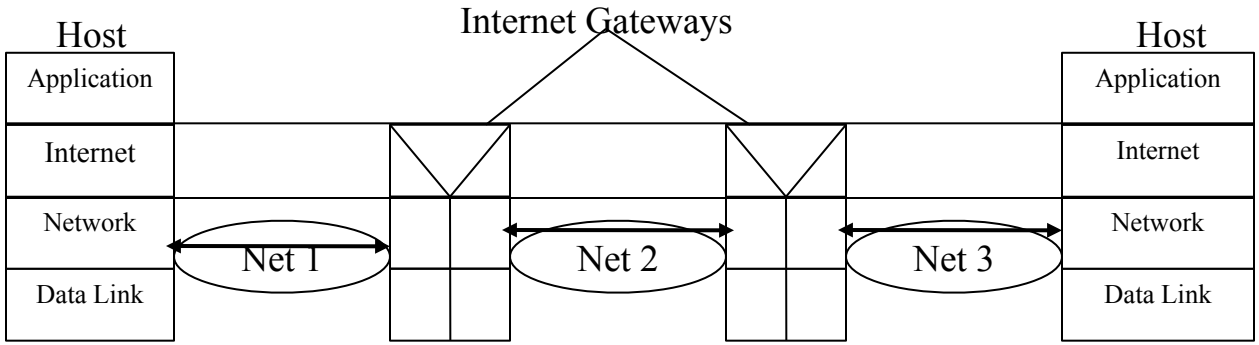
- Nope, Just Automate the Host file. Big step backwards with DNS.
- Now we have domain names
  - Macros for IP addresses
- And URLs
  - Macros for jump points in low memory

Then in ‘86: Congestion Collapse



- Caught Flat-footed. Why? Everyone knew about this?
  - Had been investigated for 15 years at that point
- With a *Network* Architecture they put it in Transport.
  - Worst place.
- Most important property of any congestion control scheme is minimizing *time to notify*. Internet maximizes it and its variance.
- And implicit detection makes it predatory.
  - Virtually impossible to fix
- Whereas,

Congestion Control in an Internet is Clearly a Network Problem



- With an Internet Architecture, it clearly goes in the Network Layer
  - Which was what everyone else had done.
- Time to Notify can be bounded and with less variance.
- Explicit Congestion Detection confines its effects to a specific network and to a specific layer.

Would be Nice to Manage the Network

- With a choice between a modern object-oriented protocol (HEMS) and a traditional approach (SNMP), err sorry . . . "a simple approach." IETF goes with "simple."
  - Must be simple has the Largest implementation of the 3:
    - SNMP, HEMS, CMIP.
  - So simple too complex to use
    - IEEE had tried the SNMP approach several years earlier so the shortcomings were well-known.
- Everything connectionless making it impossible to snapshot tables
- No attempt at commonality across MIBs.
- Router vendors played them for suckers and they fell for it.
  - Not secure, can't use for configuration.

IPv6 Still Names the Interface?

Why on Earth?

- Known about this problem since 1972
  - No Multihoming, kludged mobility
  - Notice in an Internet Architecture this is straightforward.
  - Signs the Internet crowd didn't understand the Tinker AFB lesson.
  - DARPA never made them fix it.
- By the Time of IPng, Tradition trumps Everything.
- IPv7 would have fixed it.
  - But that fight was too intense. This is no longer science, let alone engineering.
- When can't ignore, and given post-IPng trauma they look for a workaround.

Violá!

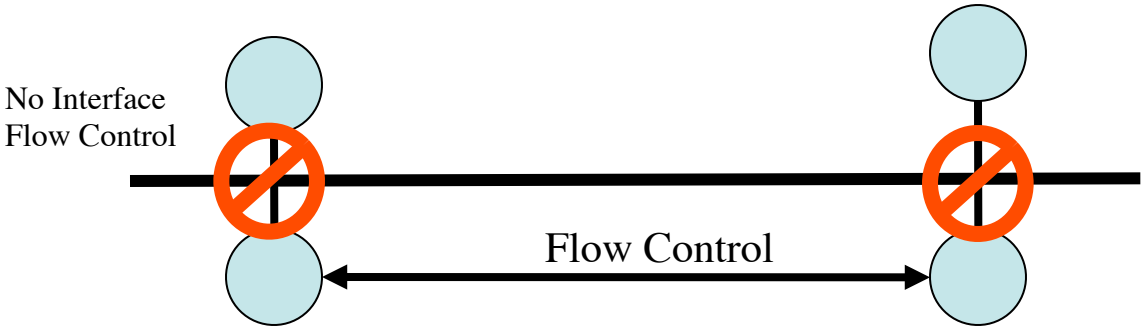
Loc/Id Split!

Loc/ID Split

(these are people who
lost a layer to begin with, right?)

- You've got to be kidding?! Right!
- First off:
  - Saltzer [1977] defines "resolve" as in "resolving a name" as "to locate an object in a particular context, given its name."
  - All names in computing locate something.
    - So either nothing can be identified without locating it, nor located without identifying it, OR
    - anything that doesn't locate something is being used outside its context
    - Hence it is either a false distinction or it is meaningless.
- Second, one must route to the *end* of the path.
  - The locator is on the path to the end, it isn't the *end*.
  - The "identifier" locates the end of the path but they aren't routing on it.
  - No wonder it doesn't scale
- There is no workaround. IP is fundamentally flawed.

Never Get a Busy Signal on the Internet!
2010 They Discovered Buffer Bloat!



- Golly Gee Whiz! What a Surprise!!
- With Plenty of Memory in NICs, Getting huge amounts of buffer space backing up behind flow control.
- If peer flow control in the protocol, pretty obvious one needs interface flow control as well.
- Well, Duh! What did you think was going to happen?
  - If you push back, it has to go somewhere!
  - Now you can have local congestion collapse!

Taking Stock

- The Internet has:
  - Botched the protocol design
  - Botched the architecture
  - Botched the naming and addressing
  - When they had an opportunity move in the right direction with application names, they didn't. They did DNS.
  - When they had an opportunity to move in the right direction with node addresses, they didn't. They did IPv6.
  - More than Botched Network Management
  - Botched the Congestion Control twice
  - Once so bad it probably can not be fixed.

- By my count this makes them 0 for 8!
- It defies reason! Do these guys have an anti-Midas touch or wha!?

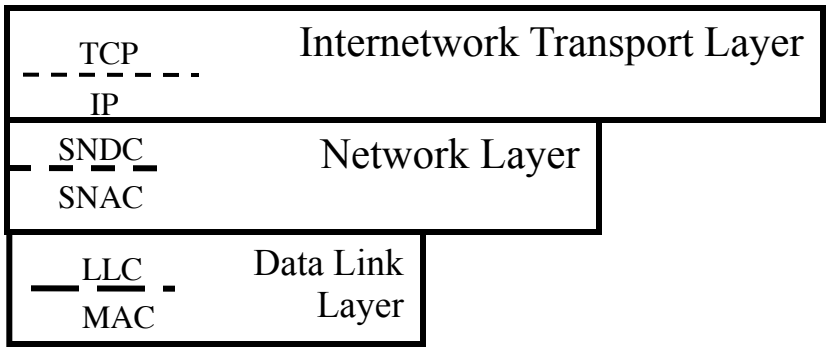
But It is a Triumph!

(By that argument, so was DOS)

- But It Works!
- So did DOS. Still does.
- With enough Thrust even Pigs Can Fly!
- As long as fiber and Moore's Law stayed ahead of Internet Growth, there was no need to confront the mistakes.

- Now it is catching up to us and can't be fixed.
  - Fundamentally flawed, a dead end.
  - Any further effort based on IP is a waste of time and effort.
    - Throwing good money after bad
  - Every patch (and that is all we see) is taking us further from where we need to be.

INWG Was on The Right Track!!



- They were Close to Seeing it was a Repeating Structure
  - A Structure we arrived at independently by a similar approach.
  - Is RINA the answer? Who knows? We are doing the science and letting the chips fall where they may.
  - Can you propose something that is simpler and answers more questions, makes predictions about things we haven't seen?
  - We can and have. . . .

How Lucky Can You Get!?

- Now that We are Back on Track, There is so much to explore!
  - It is Interesting How Different the Fundamentals Are
- By Maximizing Invariances and Minimizing Discontinuities:
  - To scale, resource allocation problems require a repeating (recursive) structure. (confirmed by Herb Simon's Science of the Artificial)
  - A Layer is a Distributed Application for Managing Interprocess Communication.
  - Watson's results are fundamental: Bounding 3 Timers are the Necessary and Sufficient conditions for synchronization.
    - Besides a simpler protocol
    - Implies decoupling port allocation and synchronization
    - Implies that for it to be networking (IPC) MPL must be bounded.
    - Has important implications for security

Really Lucky!

- So Much to Explore!
  - Multiple layers of the same rank implies a means to choose which one to use.
    - Neither a global address space nor a global name space are absolute requirements
  - The nature of security is much clearer, simpler, and more robust.
  - Many new avenues to explore in congestion control and quality of service.
  - And the adoption can be seamless.
  - And there are implications beyond IPC to distributed systems in general that we are just beginning to understand:
    - Hint: Distributed Applications are Local Computations.
- Is there more to Discover?
- Undoubtedly!