Look Before You Leap

ITU-T FG 2030 Workshop
New York October 2nd. 2018
John Day
email:jeanjour@comcast.net

Maximizing the invariances, and
Minimizing the discontinuities.
A Solid Foundation for the Future

Getting to 2030
Why is There a Problem?

- Complexity, Complexity, Complexity.
  - Problems aren't solved; they are papered over. The 'Net keeps getting more and more complex.
- Scalability, Scalability, Scalability.
  - Nothing scales, and growth is skyrocketing
- Security, Security, Security.
  - Daily headlines of millions compromised. (not good). Atlantic Council predicts that we are at the tipping point where the Internet ROI goes negative.
- Throughput and response time are becoming problems at scale.
  - No fundamental change in approach to network resource allocation in 50 years.
    - Just throw bandwidth at it.
- Damage to the social fabric.

Are the Problems Really That Bad?

- **The Data aren't optimistic.**
- **In 2001, Blumenthal and Clark write "*Rethinking the Design of the Internet: The End-to-End Argument vs the Brave New World*."**
  - Has a long list of problems for which e2e doesn't provide direction.
  - Classic characteristic of a paradigm in crisis (see Kuhn). No surprise.
- **For two decades researchers world-wide have been searching for a Future Internet Architecture. They failed.**
- **Why?**
  - Because they concentrated on what to build, rather than what they didn't understand.
    - For details, See my keynote to the Future Internet Assembly, Budapest, 2011.
- **Didn't understand? Surely not.**
  - It certainly appears that way.

But We Just Need to Fix a Few Things?

- **Right?**
  - We heard that earlier today.
- **The flaws are much more fundamental.**
  - The Internet is not an internet, or it is in the same sense that the PSTN is.
    - In 1983, they lost the Internet Layer and became a network.
  - The Internet has less than a third of required naming and addressing.
    - This impacts nearly everything from resource allocation (routing) to multihoming to mobility.
  - Security can't be retrofitted and the piecemeal attempts are adding considerable complexity and huge costs.
  - Congestion control causes congestion (and loss of data), thwarts doing anything about QoS, is predatory, and hence probably can't be fixed.
- **The solution to the addressing problems have been known since 1972 (multi-homing).**
- **A non-predatory congestion approach that would greatly reduce loss has been known since 1988.**
- **All indications are that either problems can't be fixed technically or the political will is not there. (IPv6)**
- **And to shockingly, they didn't understand layers**

The (really) Important Thing about Layers

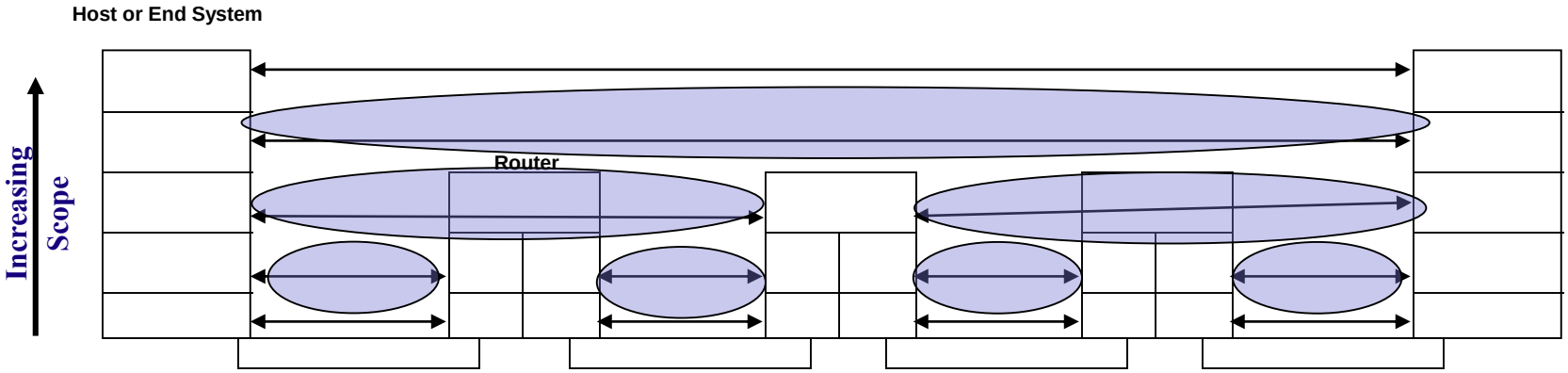
- A Layer is a locus of distributed shared state.

  In OSs, Layering was a convenience, here it is a necessity.

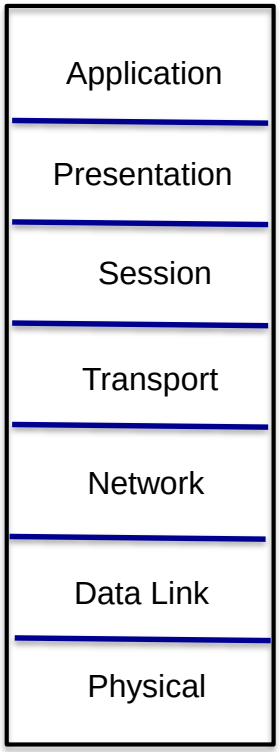
- **Different Layers should have different scope**

  Either in terms of number of elements or range of operation

  There are multiple layers at the same rank.



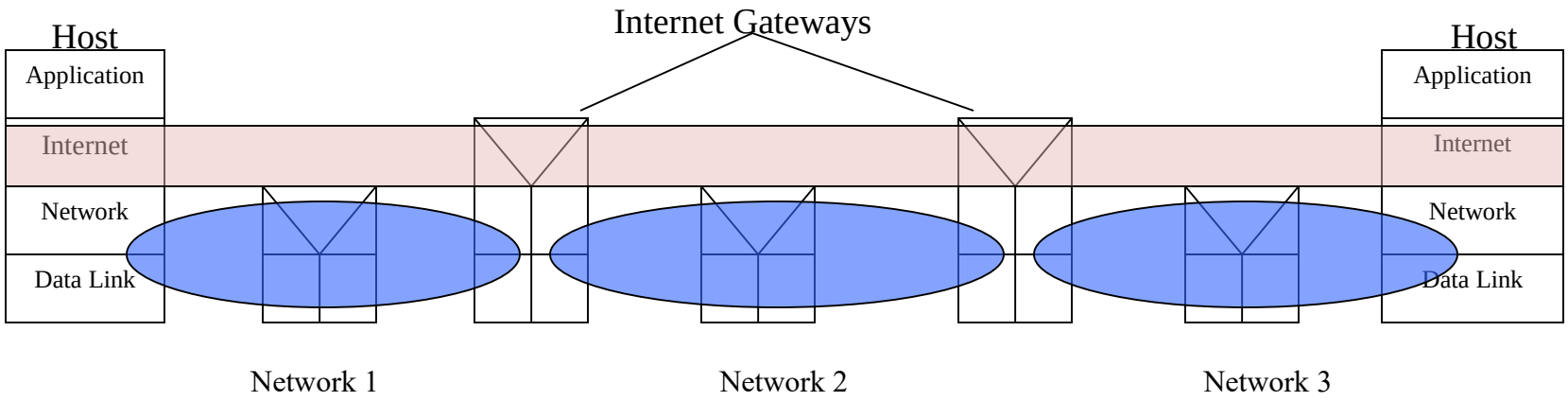
This was just a short-hand when space was tight.



- They thought this *was* the picture.
  - The emphasis on stacks

Internet Model

(circa 1975)*

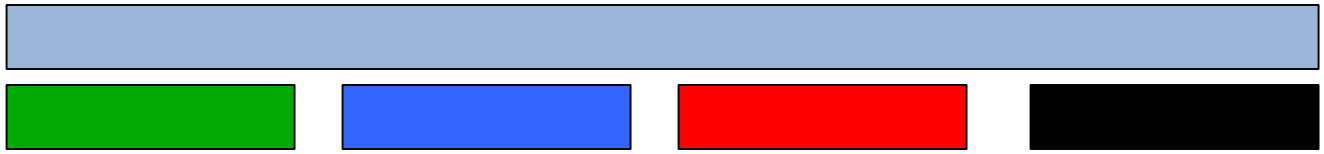


- **An Internet Layer addressed Hosts and Internet Gateways.**
- **Several Network Layers of different scope, possibly different technology, addressing hosts on that network and that network's routers and its gateways.**
  - *Inter-domain routing at the Internet Layer;*
  - *Intra-Domain routing at the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

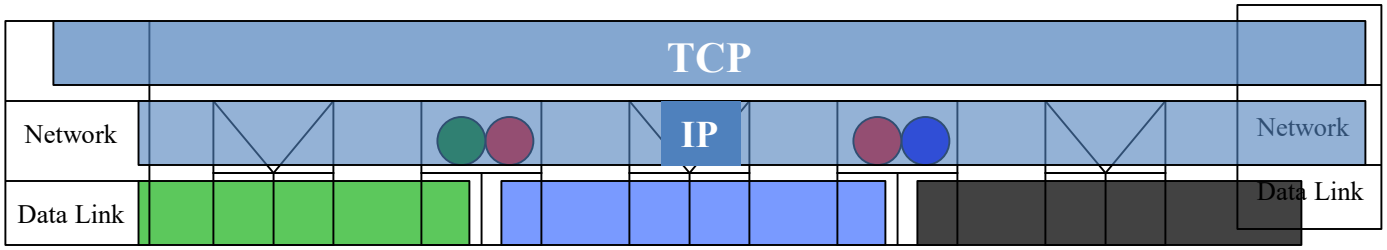
* Independently discovered in ISO 8648

How Did the Internet Lose a Layer?

- **Consider the networks for DARPA's Internet in the late 70s:**
  - Ethernet
  - Packet Radio
  - Satellite Network and of course
  - the ARPANET, but it is a black box.
- **And They are Building an INTERNET!**
  - So they need an overlay



- **But the first 3 are not networks, but multi-access media** ***(only link layers)***
  - And with the common overlay



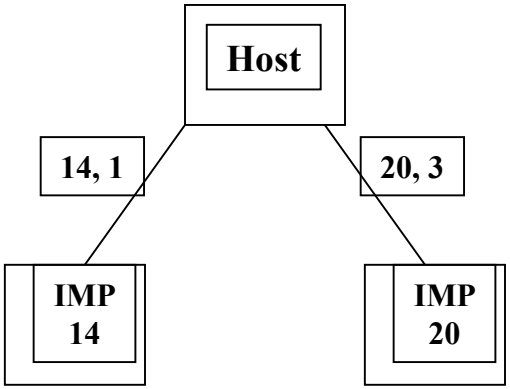
- **And then split IP from TCP. This should look familiar.**

In the ARPANET, a Host Address was an IMP port number.
Did It Take Long to Realize
It Was a Problem?

Nope.

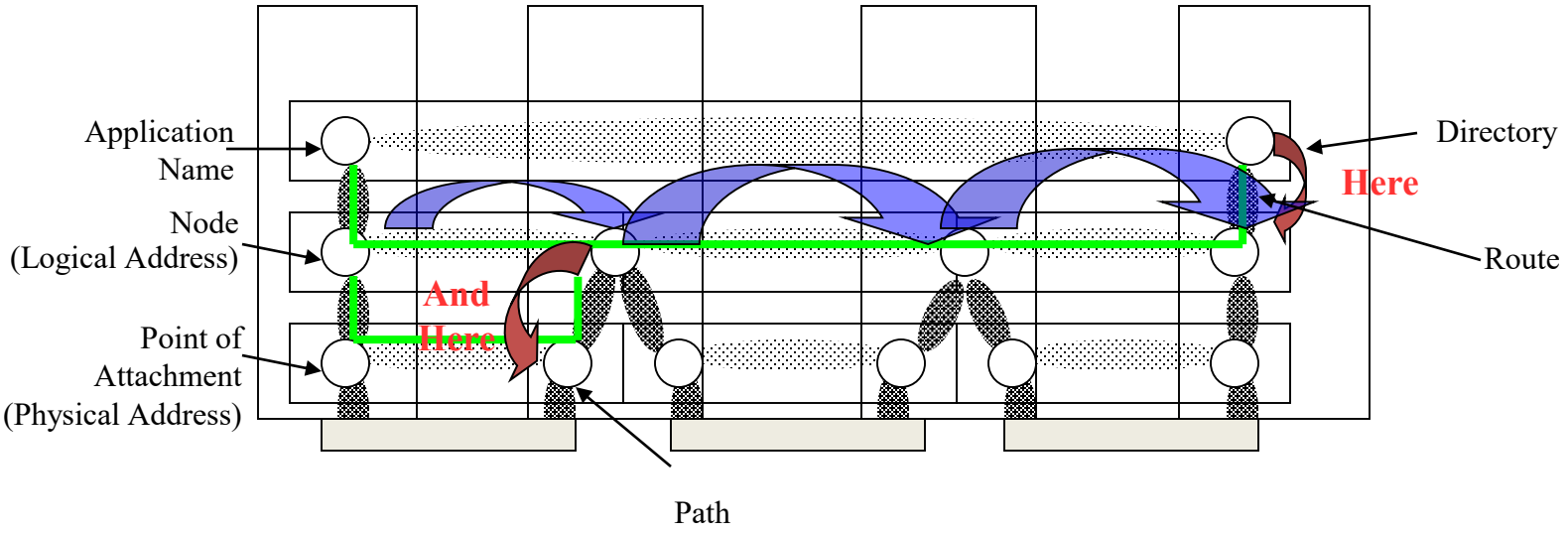
First time (~ 1972) one of the Air Force bases took us at our word that the network was suppose to be survivable and asked for links to two different IMPs to connect its host to the Network.

Naming the hosts by the names of their interfaces meant that the two connections looked like two hosts to the Net.
Still does.



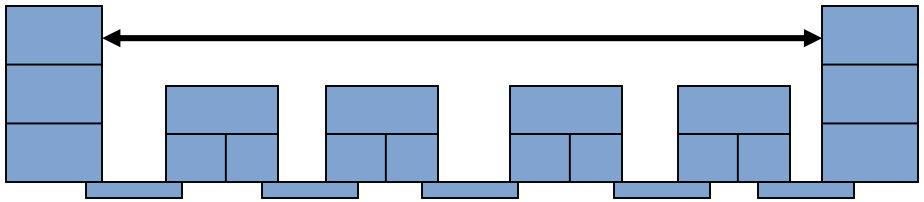
What Are The Necessary Names?

- **Remember Tinker AFB? The answer was obvious. Just like OSs!**
  - But a bit more general
- **Directory** provides the mapping between Application-Names and the node addresses of all Applications reachable without an application relay.
- **Routes** are sequences of node addresses used to compute the next hop.
- **Node to point of attachment mapping** for all nearest neighbors to choose path to next hop. (Because there can be multiple paths to the next hop.)
- **This last mapping and the Directory are the same:**
  - Mapping of a name in the layer above to a name in the layer below of all nearest neighbors.



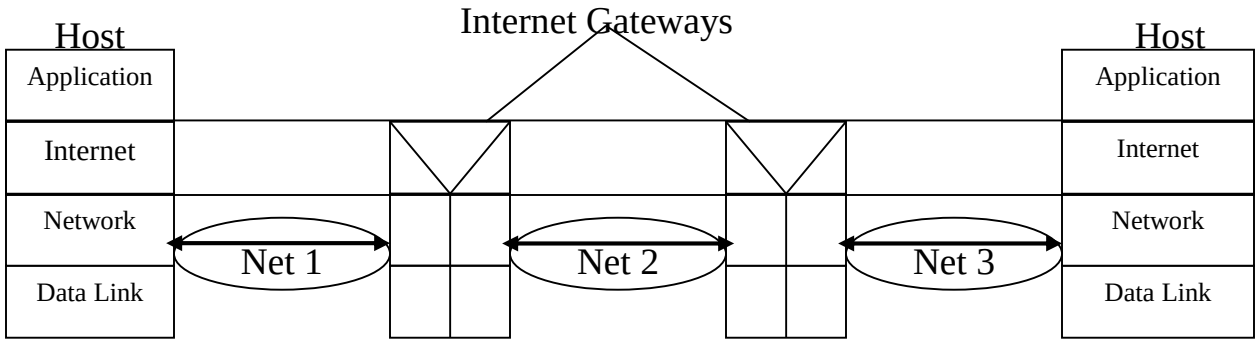
What is Wrong With Congestion Control?

What's Right!? O, they did get AIMD right, but that's all.



- **The Effectiveness of *Any* Congestion Control Strategy deteriorates with increasing time to notify.**
  - Putting it Transport maximizes time-to-notify and it increases as the network grows.
- **It works by causing congestion!**
  - It keeps pushing the cliff, rather than looking for the knee.
- **And implicit detection makes it predatory.**
  - Don't see a way out that isn't very painful
- **And it thwarts the Network Layer from doing QoS**
- **At best, this is a network solution, not an internet solution**
- **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 thought.
  - Where it can be coordinated with QoS mechanisms.
- **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.**
  - Did we know better at the time? Yes, Raj Jain's work

So We *Do* Need a New Paradigm?

- **The Good News is Not So Much '*New*,'**
- **We already have the 'new' paradigm and have had for 45 years.**
- **The insights developed primarily by the CYCLADES group.**
  - You mean, Datagrams and end-to-end Transport?
  - Well, that too, but that was just tip of the iceberg, the most visible aspect.
- **Pouzin's team was on the trail of a new paradigm.**
  - Remember, the ARPANET was built to be a production network
  - CYCLADES was built to be a network to do research on networks
    - They weren't the only ones: Xerox, Lawrence Livermore, etc.
    - There was a general idea that as Bob Metcalfe mentioned in passing in 1972:
    - Networking is Interprocess Communication (IPC)

The New Model Had 4 Characteristics

- **It was a *peer network* of communicating equals** not a hierarchical network connecting a mainframe master with terminal slaves.
- **The approach required coordinating *distributed shared state at different scopes*, which were treated as black boxes.** This led to the concept of layers being adopted from operating systems and….
- **There was a shift from largely deterministic to *non-deterministic* approaches**, not just with datagrams in networks, but also with interrupt driven, as opposed to polled, operating systems, and physical media like Ethernet, and last but far from least…..
- **This was a computing model, *a distributed computing model*, not a Telecom or Data comm model.**
  - Notice how the Internet (and all the diagrams today) emphasize boxes, when they should be emphasizing layers, as a Distributed Application that does IPC.

Picking up the Threads

- By the 90s, it was clear that we had gone wrong someplace.
- The 7-layer model turned out not to be what we thought:
  - By 1983, we knew the upper 3 layers were one layer.
  - By 1986, OSI figured out independently what INWG knew in 1975 that there was a network layer and an internet layer.
- Determined I needed to find out what I *knew* about networks independent of hardware, standards, politics, etc.
  - There had been some promising patterns that had come up during the OSI work. But there was more interest in holding to a political position than finding out what the problem said.
  - Separating mechanism and policy
  - The lower layers were looking very similar.
    - But then I can't tell a coffee cup from a donut.
  - And Dick Watson's seminal result on protocols:

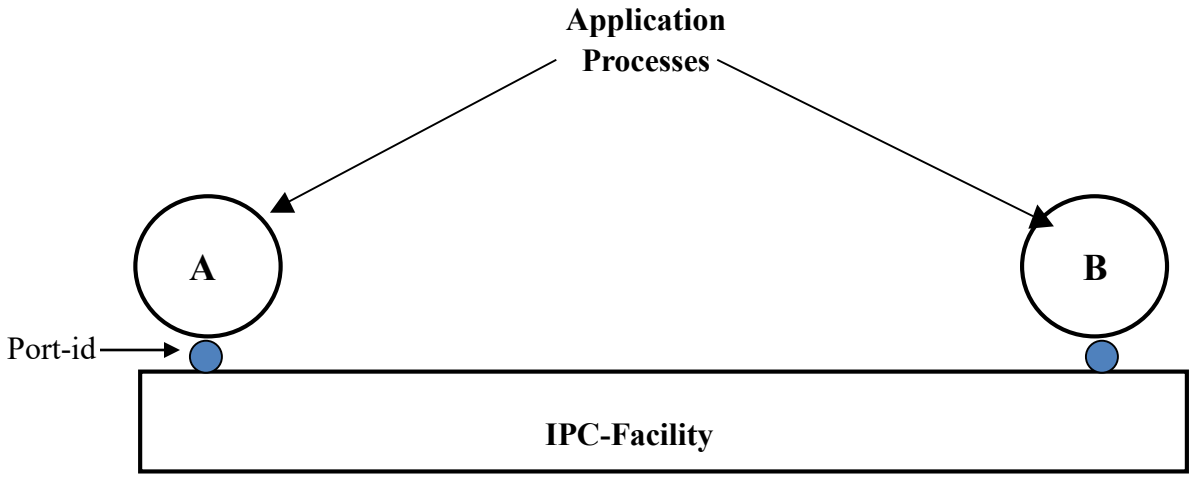
Necessary and Sufficient

- In the late 1970s, Richard Watson makes a remarkable discovery.
- **The *necessary and sufficient* conditions for Synchronization of Reliable Data Transfer is to bound three timers:**
  - MPL - Maximum Packet Lifetime
  - A - Maximum Hold Time before Ack
  - R - Maximum Time to Re-try
- Assumes all connections exist all the time and always have
- A state vector is merely a cache of information on recently active connections.
- **No traffic for 2(MPL+A+R), discard the state vector.**
  - 3 PDUs *are* Exchanged (lots more than that), 3-way handshake is irrelevant.
  - It works Because, the Timers are Bounded not because 3 PDUs were exchanged
- **This yields a more robust and more secure protocol.**
- **But even more fundamental is:**
  - If MPL is bounded it is networking, If not, it is a remote storage.

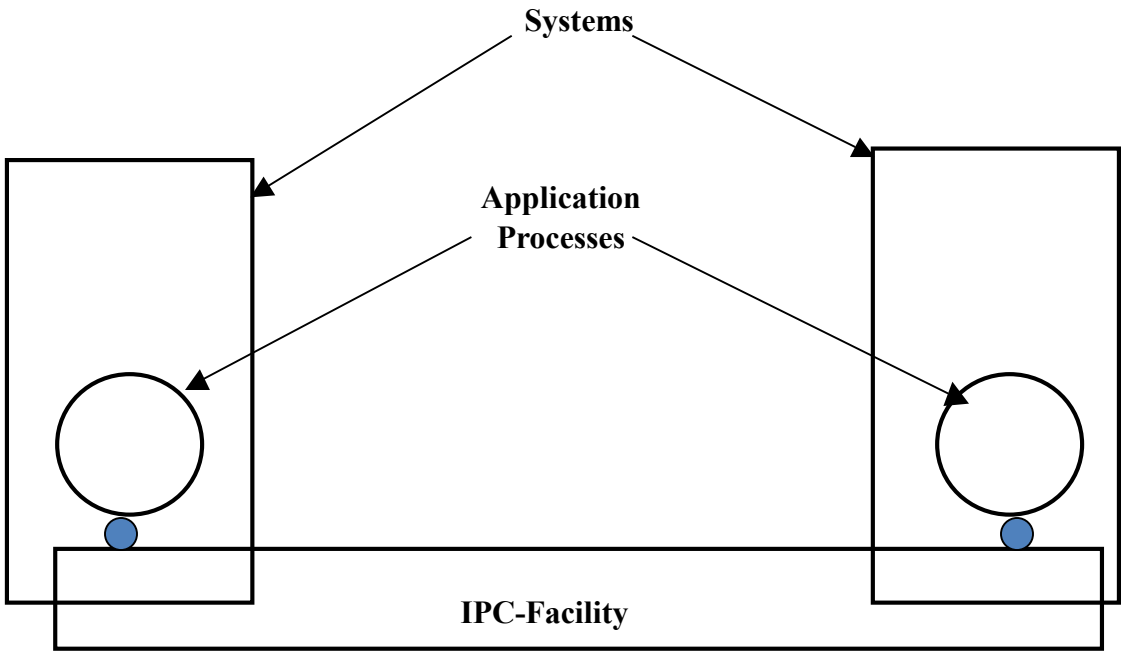
TCP bounds MPL explicitly but assumes the other two are bounded, consequently isn't as robust.

1: Start with the Basics

Two applications communicating in the same system.



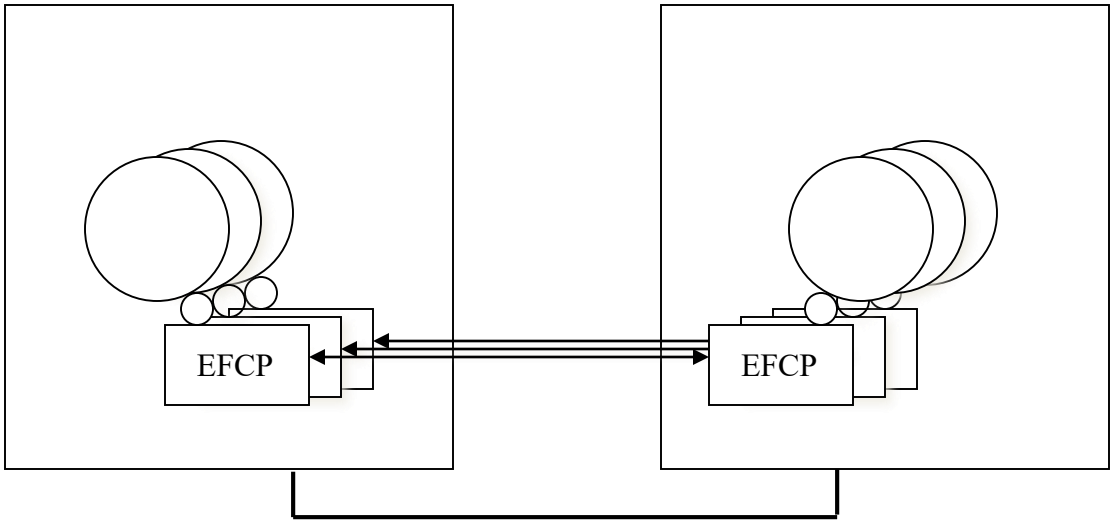
2: Two Application Communicating in Distinct Systems



3: Simultaneous Communication Between Two Systems

i.e. multiple applications at the same time

- **To support this we have multiple instances of the EFCP.**



Will have to add the ability in EFCP to distinguish one flow from another.

Connection-id

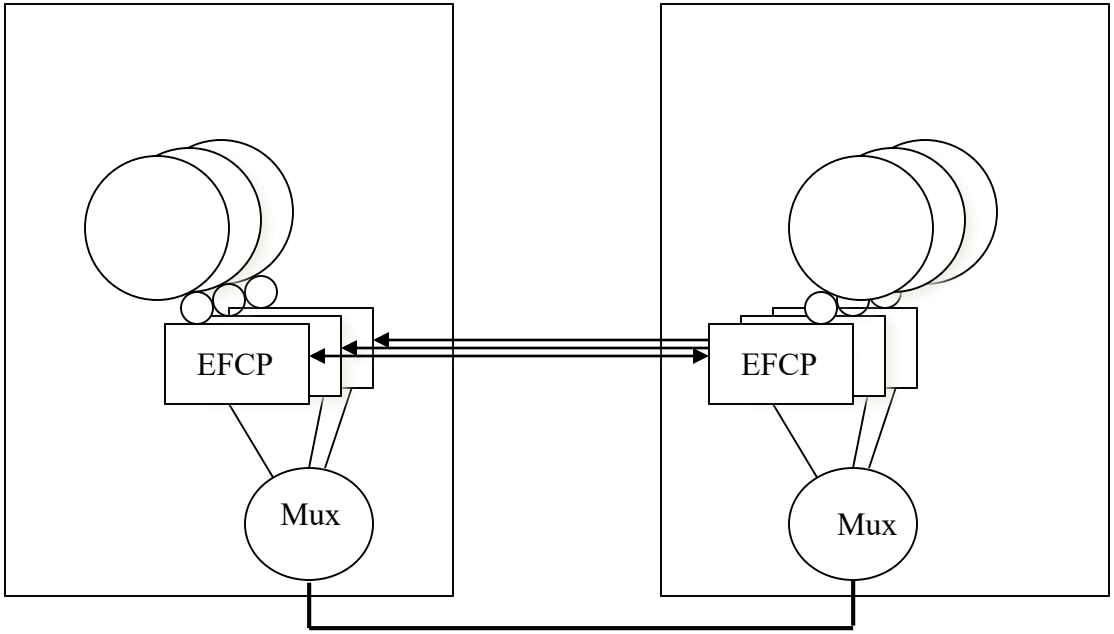
Dest-port	Src-port	Op	Seq #	CRC	Data

Typically use the port-ids of the source and destination.
Also include the port-ids in the information sent in IAP to be used in EFCP synchronization (establishment).

Simultaneous Communication Between Two Systems

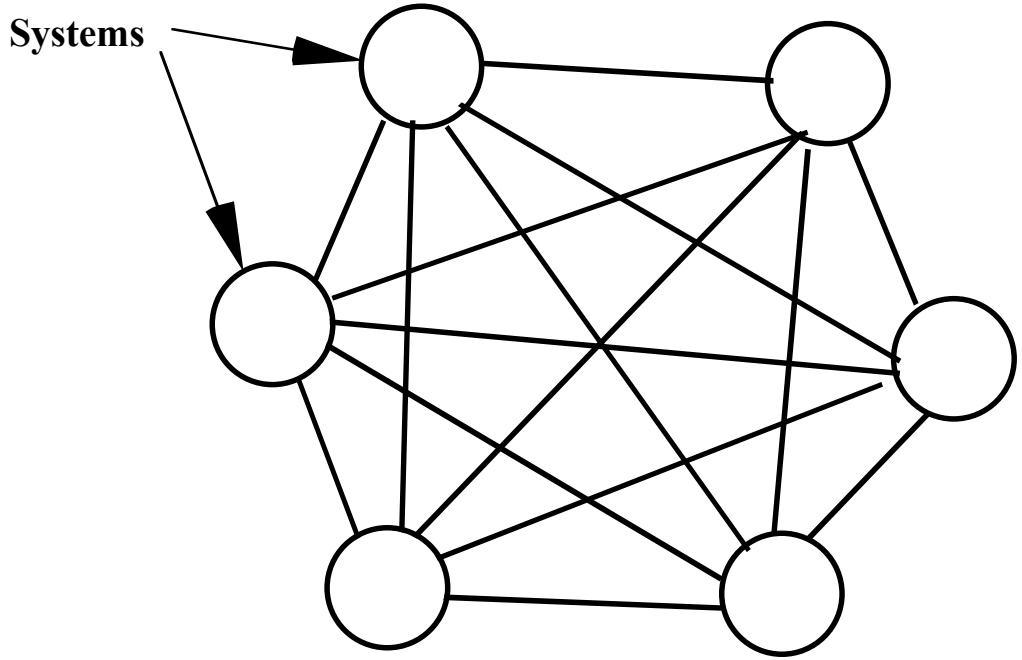
i.e. multiple applications at the same time

- **In addition to multiple instances of the EFCP.**

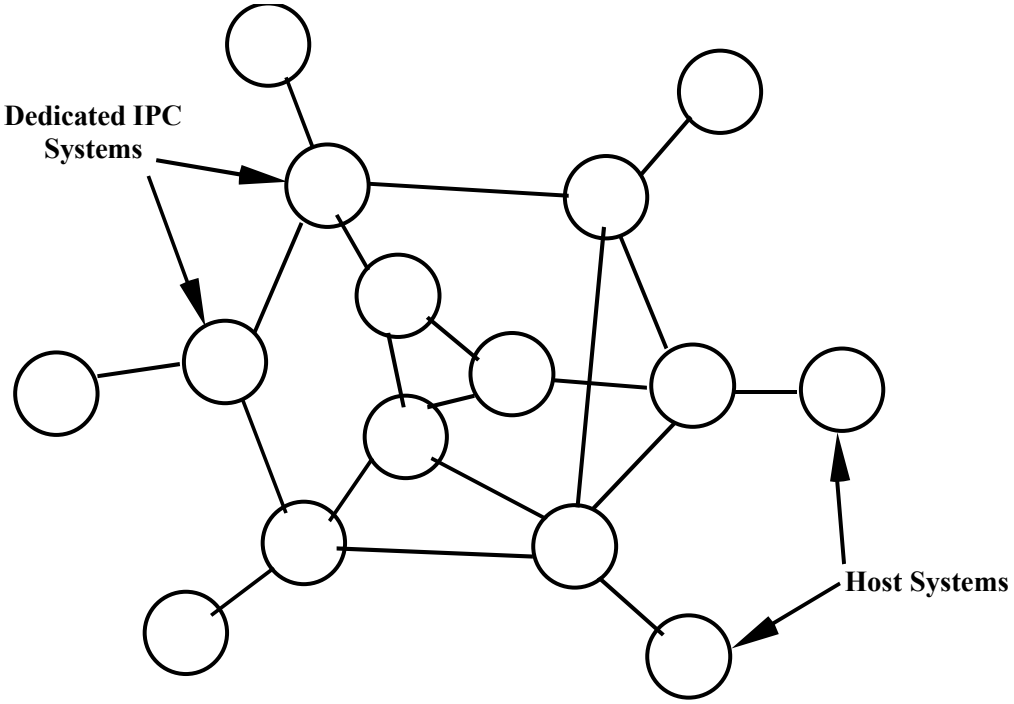


Will also need an application to manage multiple users of a single resource.

4: Communication with N Systems



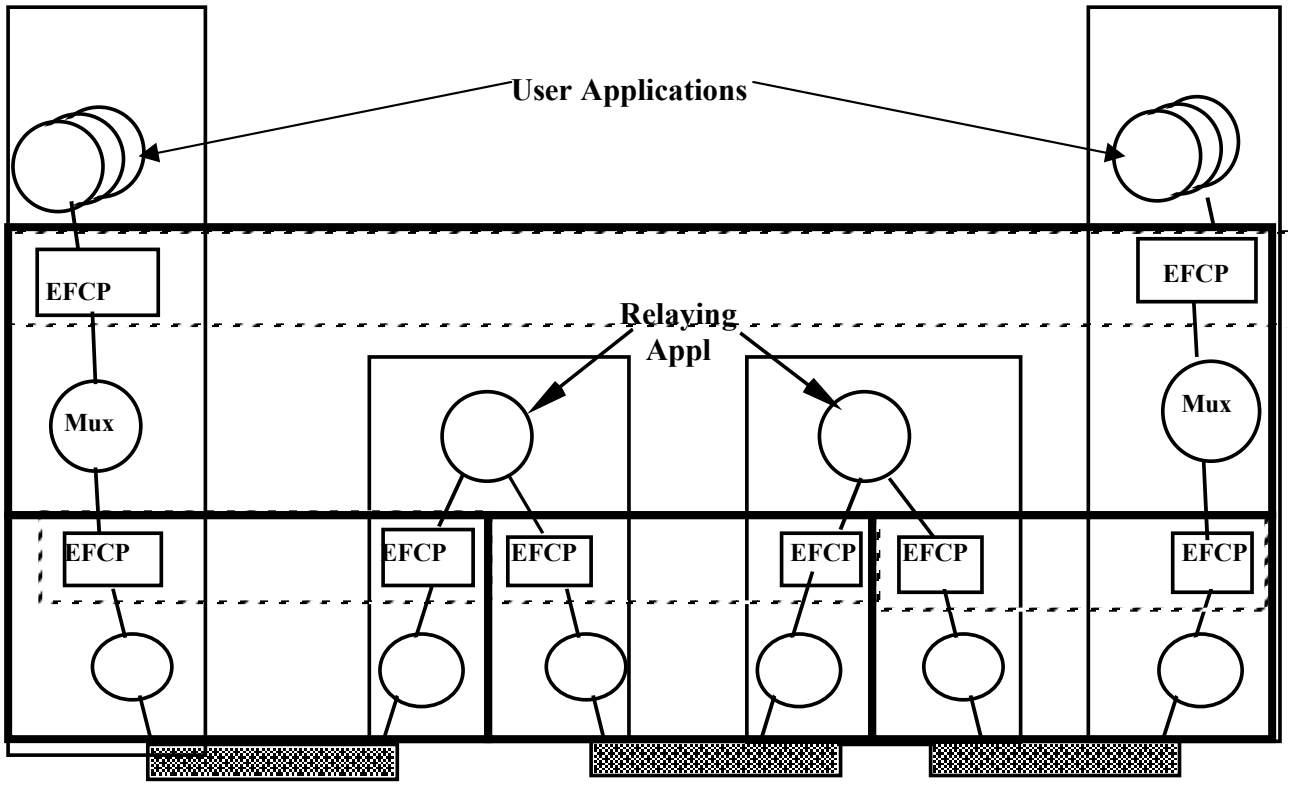
5: Communicating with N Systems (On the Cheap)



By dedicating systems to IPC, reduce the number of lines required and even out usage by recognizing that not everyone talks to everyone else the same amount.

The IPC Model

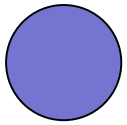
(A Purely CS View)



Distributed IPC Facilities

Software Engineering Question

- **There is this module:**



- **There is a need to partition it into two modules. The question is along what lines does one partition it?**
- **One could partition it one way, but it breaks an internal function:**

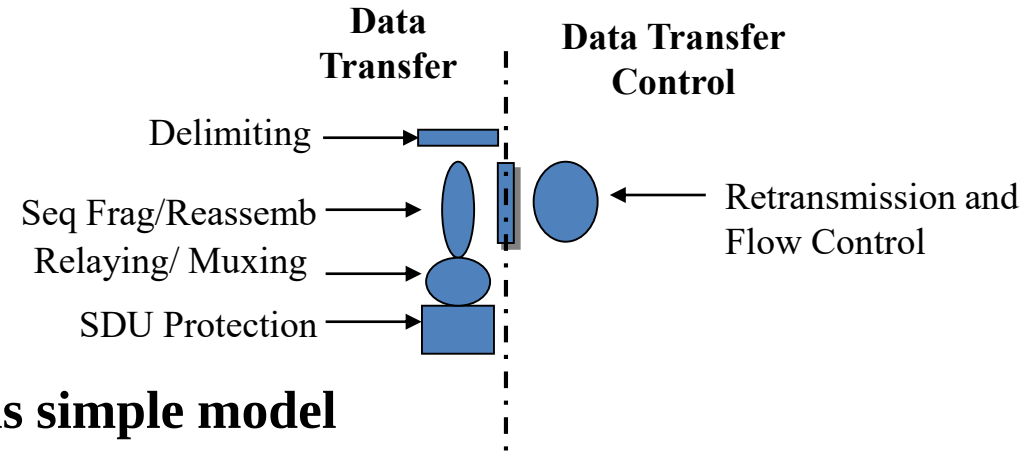
- **Or along different lines, which doesn't break anything,**

- **Which one do you choose?**

- **Then Why Didn't They Do that with TCP?!**

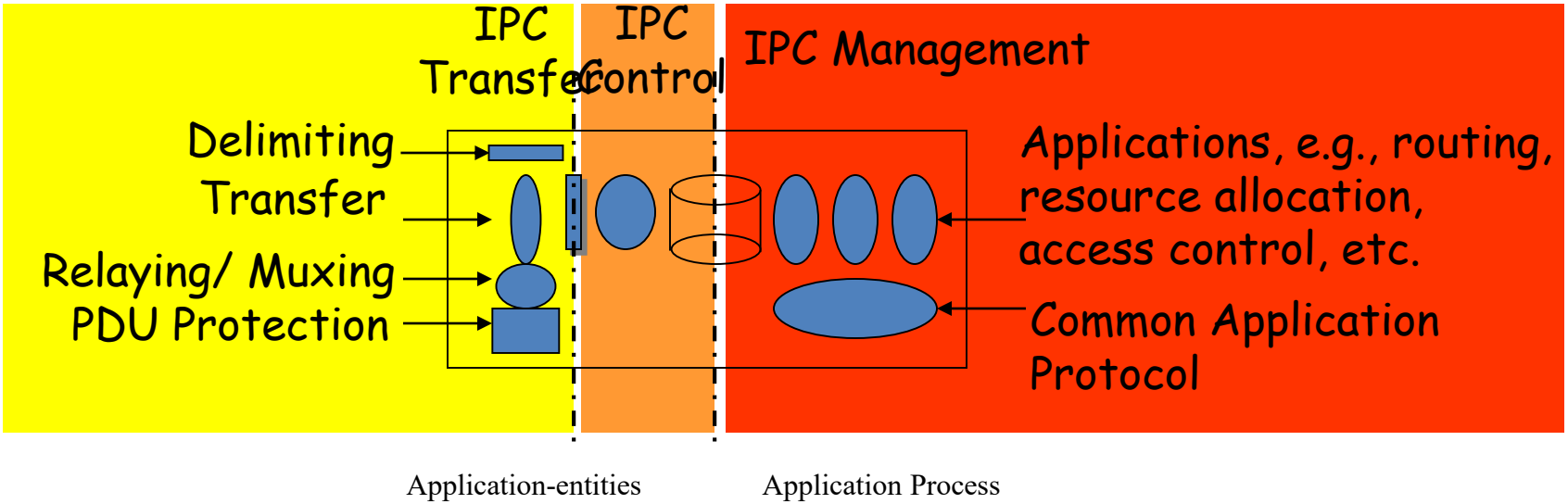
This leads to

- **Networking is IPC and only IPC.** (Well, we knew that.)
  - We Call it , RINA, but these are the fundamental principles.
- **Layers are not divisions of functionality, but divisions of allocation.**
  - All Layers have the same functions, they just operate on a different ranges of capacity, QoS, and scope.
- **Application-names and "port-ids" are the only externally visible identifiers.**
  - Applications never see addresses. Port-ids have only local significance.
  - Separating Mechanism and Policy yielded an interesting result.



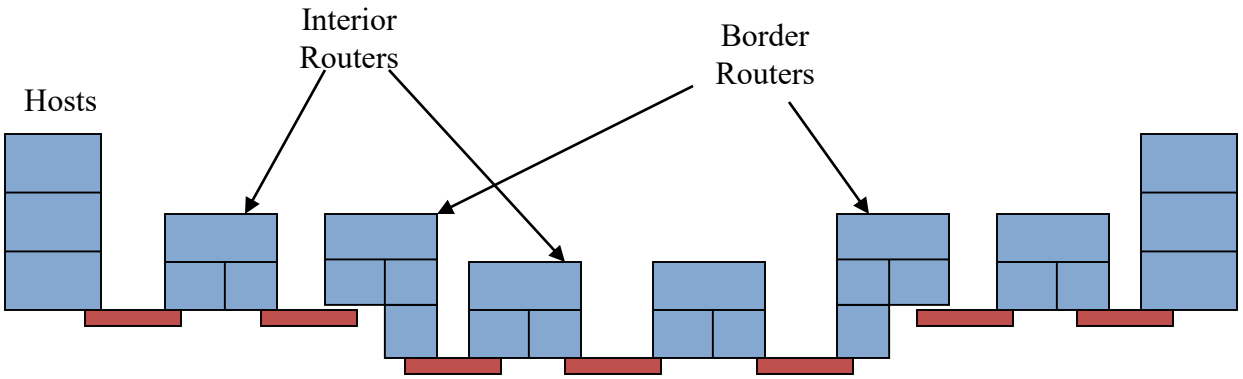
- **From this simple model**

What a Layer Looks Like



- Processing at 3 timescales, decoupled by either a State Vector or a Resource Information Base
  - IPC Transfer actually moves the data (≈ IP + UDP)
  - IPC Control (optional) for retransmission (ack) and flow control, etc.
  - IPC Layer Management for routing, resource allocation, locating applications, access control, monitoring lower layer, etc.
- Remember that within a scope if there is a partitioning of functions, it will be orthogonal? Well, here it is.

Only Three Kinds of Systems



- Middleboxes? We don't need no stinking middleboxes!
- NATs: either no where or everywhere,
  - NATs only break broken architectures
- The *Architecture* may have more layers, but no *box* need have more than the usual complement.
  - Hosts may have more layers, depending on what they do.

There are Many Incredible Results: I

- A huge complexity collapse of two orders of magnitude
- There is one layer and it repeats over different ranges of bandwidth, QoS, and Scope. This architecture scales indefinitely, only limited by physics.
  - How Many? As many as you need.
- Only one data transfer protocol and one application protocol necessary.
- Because there is a full addressing model, multihoming is inherent in the model. No additional mechanisms or cost is necessary.
- Because mobility is just multihoming where points of attachment change more frequently, mobility requires nothing new.
  - No home agents, no foreign agents, no tunnels, no anchors, no new protocols.
  - Repeating layers are used to provide quick response and scaling.
- Because addresses are synonyms a network can be renumbered in seconds to minutes without losing affecting traffic.
  - Can renumber two or more simultaneously without affecting traffic.

There are Many Incredible Results: II

- Did I mention it scales.
- Congestion Avoidance can be coordinated with QoS policies.
- Guaranteed QoS source to destination
- Physical Network Resources become technology independent to the applications. Apps not tied to specific technology, only to QoS
- Not routing on the interface reduces router table size by a factor at least 4 or 5.
- With repeating layers, Router Table Size can be bounded also increases responsiveness to changes in load and to failures.
- Much faster recovery time from failures (~13ms), can use scope of layers to make it smaller.

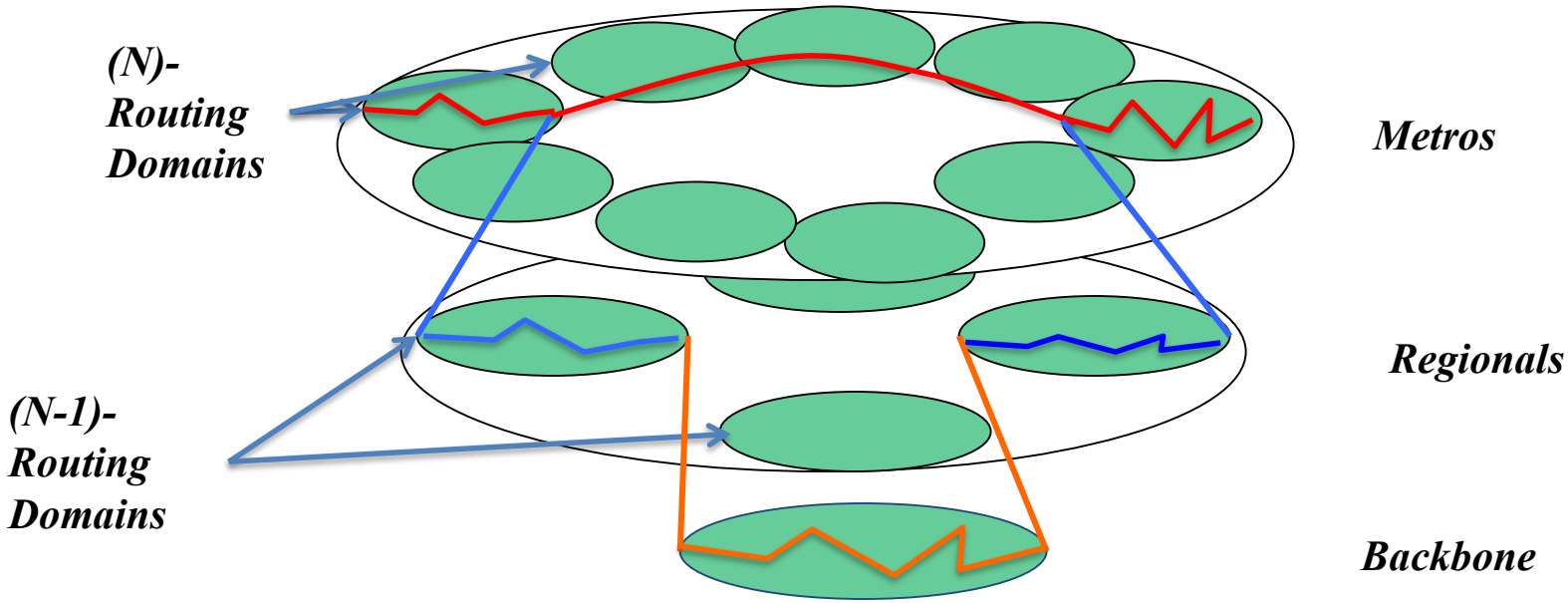
There are Many Incredible Results: III

- A Huge Complexity Collapse. Did I mention that?
- The layer is a securable container and the security is much less expensive. No firewalls are necessary.
- All networks are private. Public networks are more open private networks.
- A global address space is unnecessary. 32 or 48 bits are probably enough.
- An Application can open a connection to a specific instance of the destination application.
- Two Applications using different protocols can connect to the same instance of a destination application.

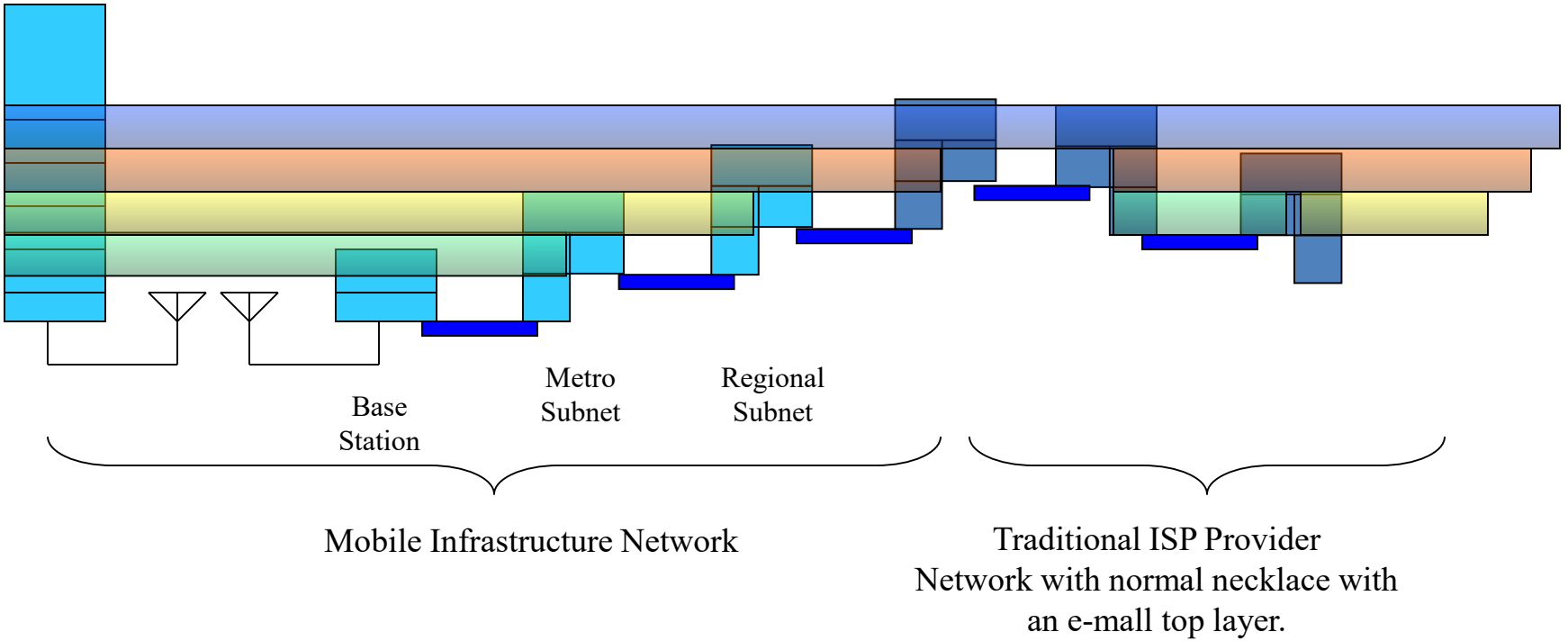
All of this has been confirmed.
We 'just ain't whistling Dixie.'

This Bounds Router Table Size

- There will be Natural Subnets *within a layer* around the Central Hole.
- Each can be a routing domain; Each Subnet is one hop across the Hole.
  - The hole is crossed in the layer below.



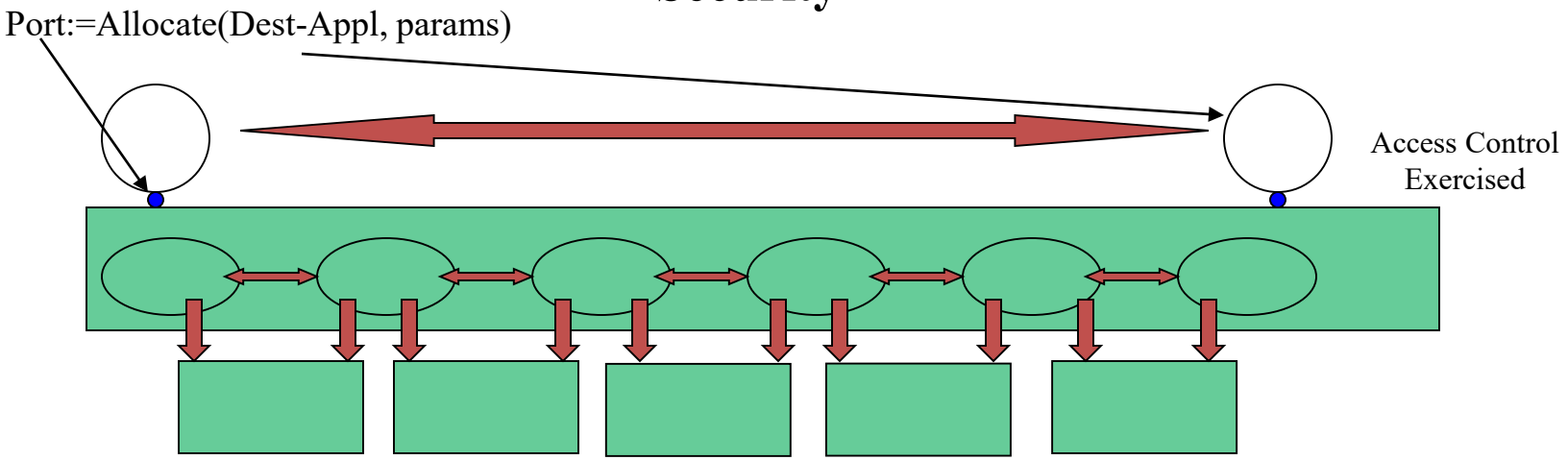
The Skewed Necklace
(Layer view)



- **Notice: No special mobility protocols. No Home Routers, No Foreign Routers, No Tunnels to set up, No special protocols. It just works.**
- **Clearly more layers could be used to ensure the scope allows sufficient time for updating relative to the time to cross the scope of the layer.**
  - Space does not permit drawing full networks.

How Does It Work?

Security



- Do What the Model Tell Us:
- Application only knows Destination Application name and its local port.
- The layer ensures that Source has access to the Destination
  - *Application must ensure Destination is who it purports to be*.
- All members of the layer are authenticated within policy.
- Minimal trust: Only that the lower layer will deliver something to someone.
- PDU Protection can provide protection from eavesdropping, etc.
  - *Complete architecture does not require a security connection, a la IPsec.*
- The DIF is a securable container. DIF is secured not each component separately.

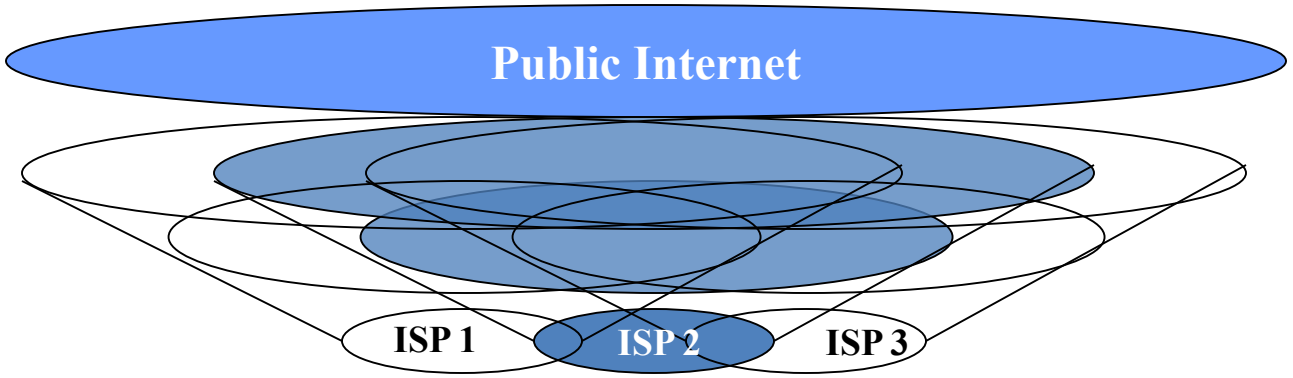
Recursive Internet Architecture (RINA) adds Inherent Security without new Protocols or new Non-Security related mechanisms

Required Functions to Add Security	Internet	RINA
Protocols	8	0
Non-Security Mechanisms	59	0
Security Mechanisms	28	7

How Does It Work?

The Internet and ISPs

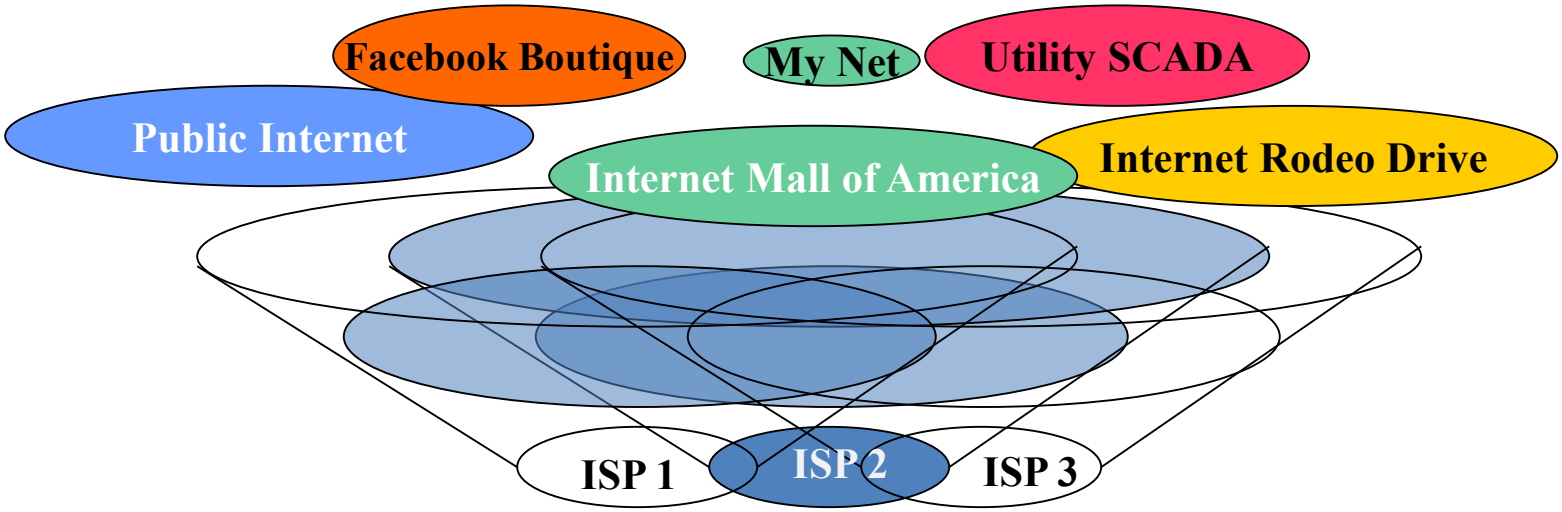
- The Internet floats on top of ISPs, an electronic-mall - “e-mall”
  - One in the seedy part of town, but an “e-mall”
  - Not the only emall and not one you always have to be connected to.



How Does It Work?

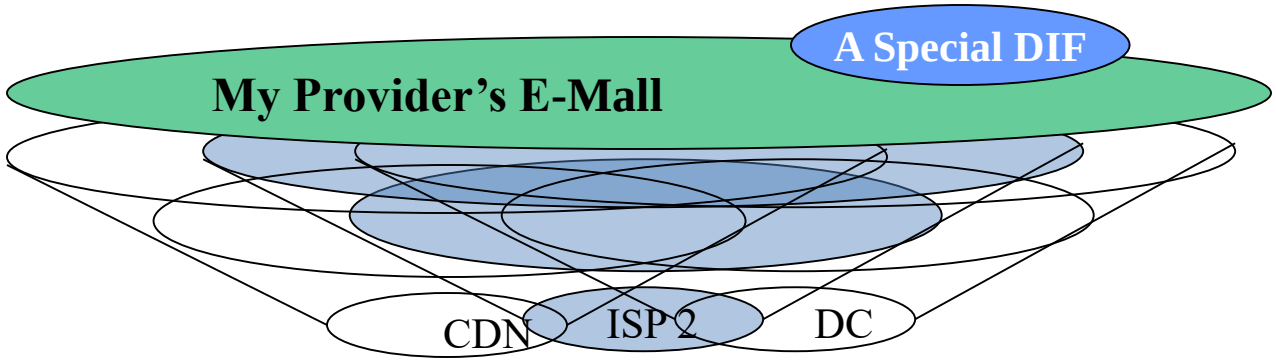
The Internet and ISPs

- But there does not need to be ONE e-mall.
  - Notice all the layers are private. Public layers are a form of private.



This Won't Make Some Happy
A Wall-less Garden?

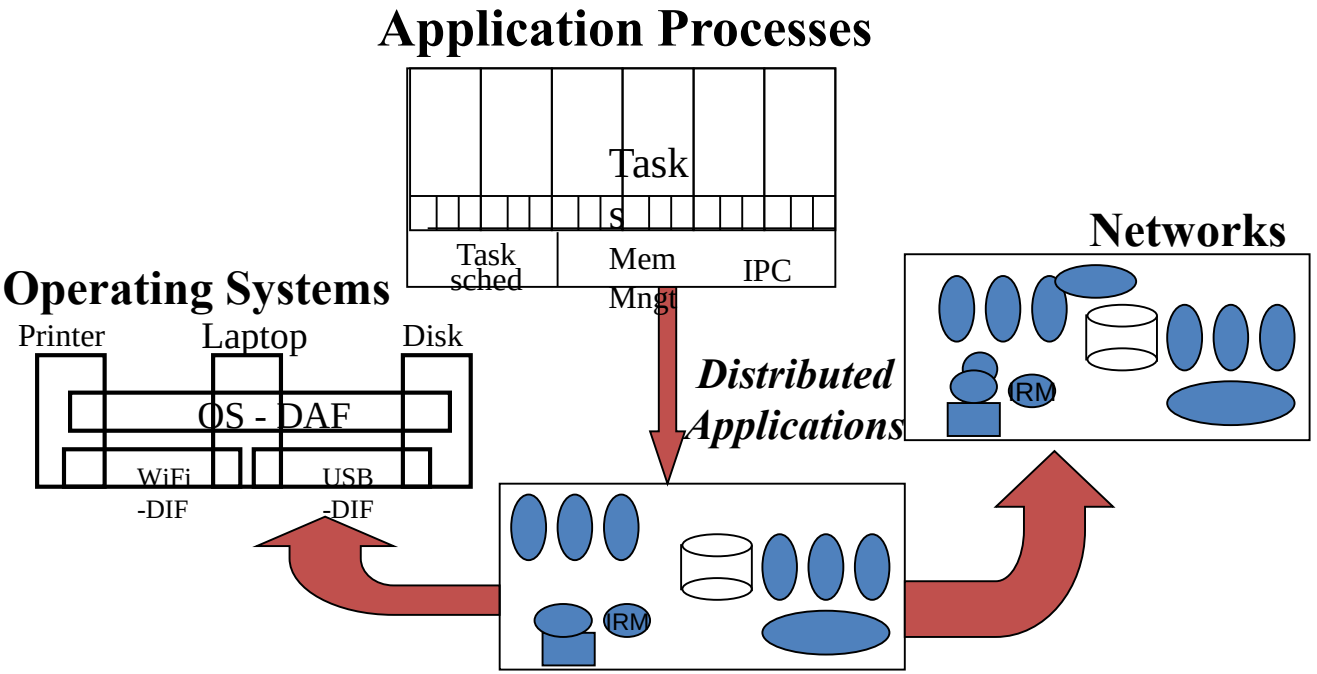
- Suppose an ISP has its own e-mall and
- forms an alliance with a few CDNs and Data Centers,
- To give the ISP access to ~80% of the most popular destinations within its e-mall.
- For the rest, create a new Special DIF for customer.
- Among other things notice the implication for security:
- An attack has to come either from:
  - » *The Customer's Network*
  - » *The ISP,*
  - » *CDN or Data Center or*
  - » *A Special DIF.*
- An "Internet" is a Non-Sequitur



Not Just a Network Model

- **A Layer is a Distributed Application that Does IPC**
- **That Forced us to Answer:**

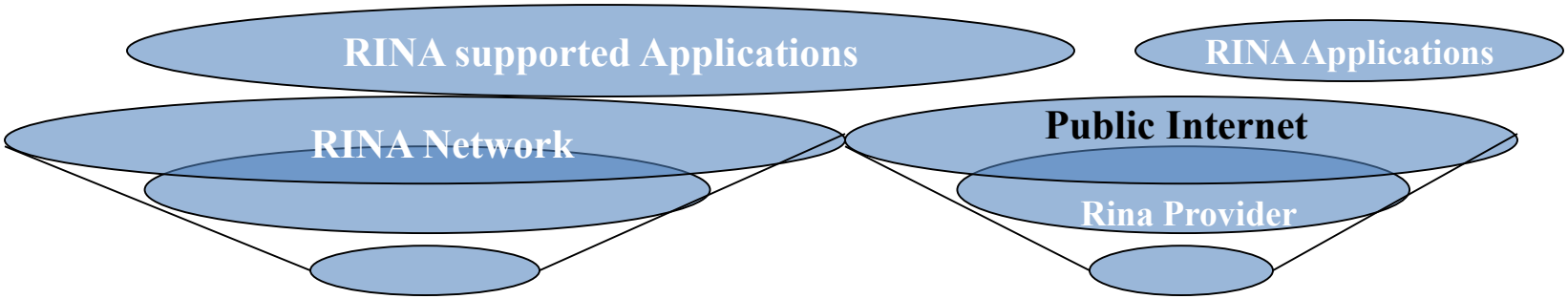
  What is a Distributed Application?
- **We now are working with a Unified Model for**



"But You Can't Replace the WHOLE Internet!"

- **Wish I had a dollar for every time I have heard *that*!**
  - What are they putting in the water these days?
- **They told *us* we would never replace the PSTN or IBM's SNA.**
  - Even in the late 1980s, people said data would never exceed voice. (!!)
- **Of course, it won't be replaced overnight. Perhaps never. Does it matter?**
  - You have already seen the transition plan.
- **Use RINA for what it is good for, Use the Internet for what it is good for:**
  - A good place to test malware, conduct cyberwarfare, steal credit cards, find drug dealers, sacrifice your privacy, etc. . . . . All sorts of useful things!
- **We build over it, under it, around it. Use it for what you want.**
- **We build other e-malls along side it.**
  - Give people a choice, after all competition is good, right?

Modify the Internet? No.
Create New and Interoperate



- **Adopt and Interoperate. Don't Modify**
  - If the old stuff is okay in the Internet e-mall, leave it there.
  - Do the new capabilities in RINA
- **Operate RINA over, under, around and through the Internet.**
  - The Internet can't be fixed, but it will run better over RINA.
  - New applications and new e-malls will be better without the legacy and run better along side or over the Internet.
- The Microsoft Approach or the Apple approach?
- A clean break with the past.
- RINA solves problems, it doesn't just fix them for now.

RINA implementations and tools

(open source)

- **IRATI**. C/C++ RINA prototype for Linux O/S, with SDK to facilitate policy development. Supports RINA over TCP, UDP, Ethernet, WiFi, shared memory; IP over RINA.
- **rlite.** Lightweight C/C++ RINA prototype, with minimality, performance and stability as design objectives. Provides same native application API as IRATI, with similar features.
- **ProtoRINA.** User-space Java implementation of RINA, with a focus on quick prototyping and teaching.
- **RINASim.** The official OMNeT++ framework for simulating RINA networks. It allows researchers to test and validate policies and RINA configurations; and facilitates visualisation and understanding of RINA principles.
- **rumba.** Python framework that allows users to define, script, execute and monitor RINA experiments on multiple testbeds (GENI, FIRE+, local), using multiple RINA prototypes (IRATI, rlite)

There is Much More, And Much More to Discover!

- An Invitation: Come explore it with us.
  - There is much to explore:
    - Believe it or not, this talk has left out a lot!
    - How it applies to different environments, especially wireless.
    - What are the dynamic properties?
      - Routing, congestion control
- Start with **Patterns in Network Architecture**, Prentice Hall
  - Then the "Reference Model" (4 sections) and
  - Check out related work at
  - At **www.pouzinsociety.org** or **ict-pristine.eu**
  - **www.irati.eu** or **ict-arcfire.eu**
  - **csr.bu.edu/rina**