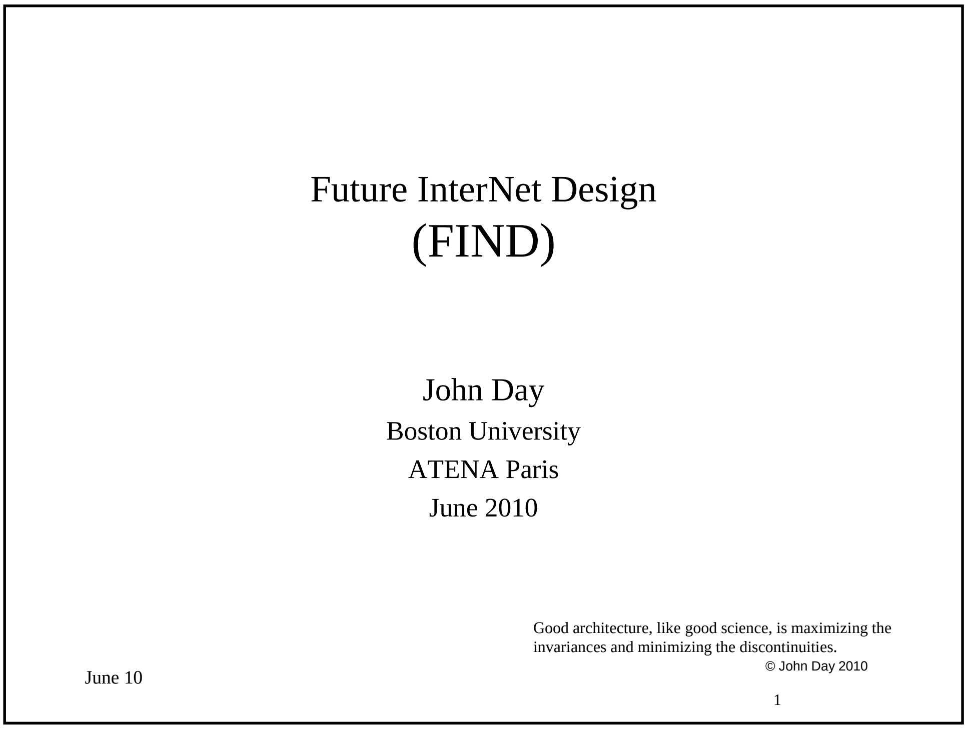
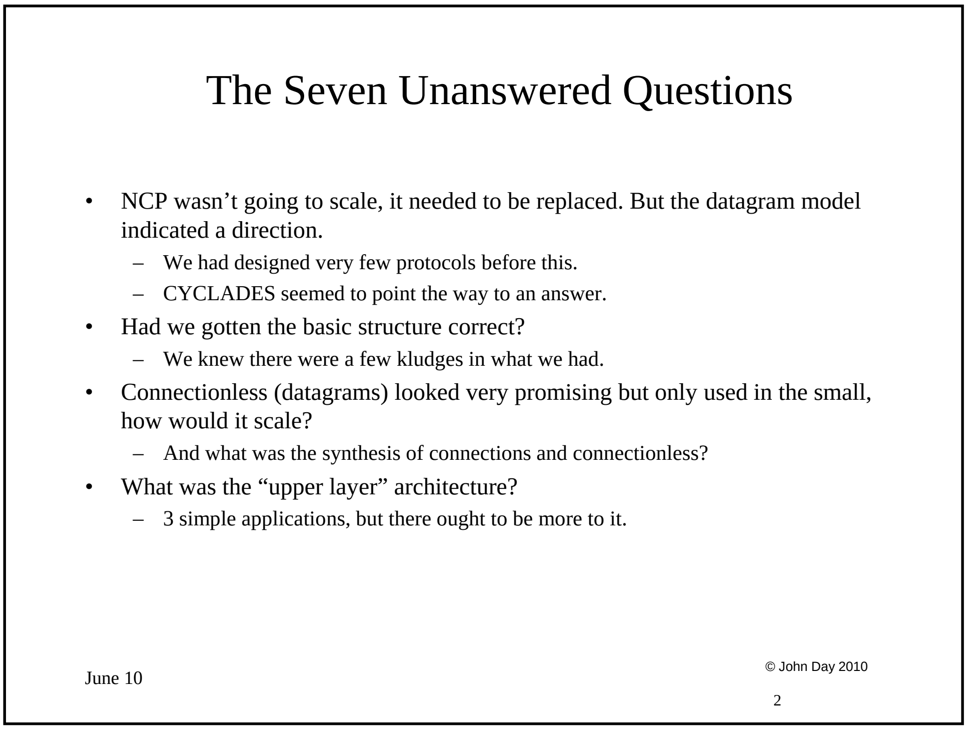


Future InterNet Design (FIND)

John Day
Boston University
ATENA Paris
June 2010

Good architecture, like good science, is maximizing the invariances and minimizing the discontinuities.

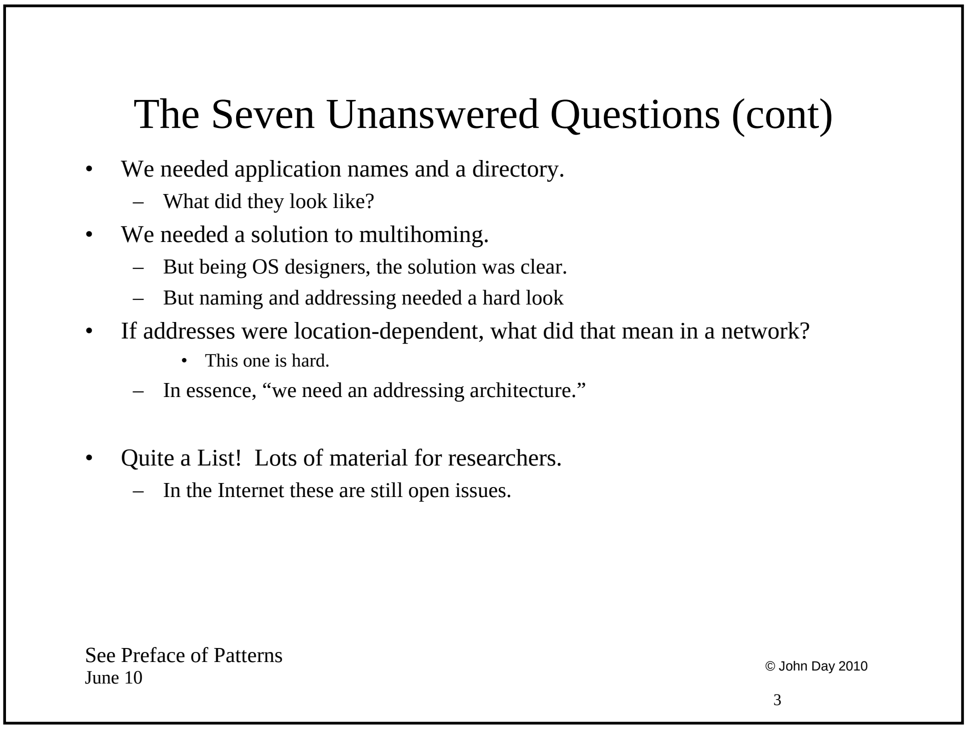
© John Day 2010



The Seven Unanswered Questions

- NCP wasn’t going to scale, it needed to be replaced. But the datagram model indicated a direction.
  - We had designed very few protocols before this.
  - CYCLADES seemed to point the way to an answer.
- Had we gotten the basic structure correct?
  - We knew there were a few kludges in what we had.
- Connectionless (datagrams) looked very promising but only used in the small, how would it scale?
  - And what was the synthesis of connections and connectionless?
- What was the “upper layer” architecture?
  - 3 simple applications, but there ought to be more to it.

© John Day 2010

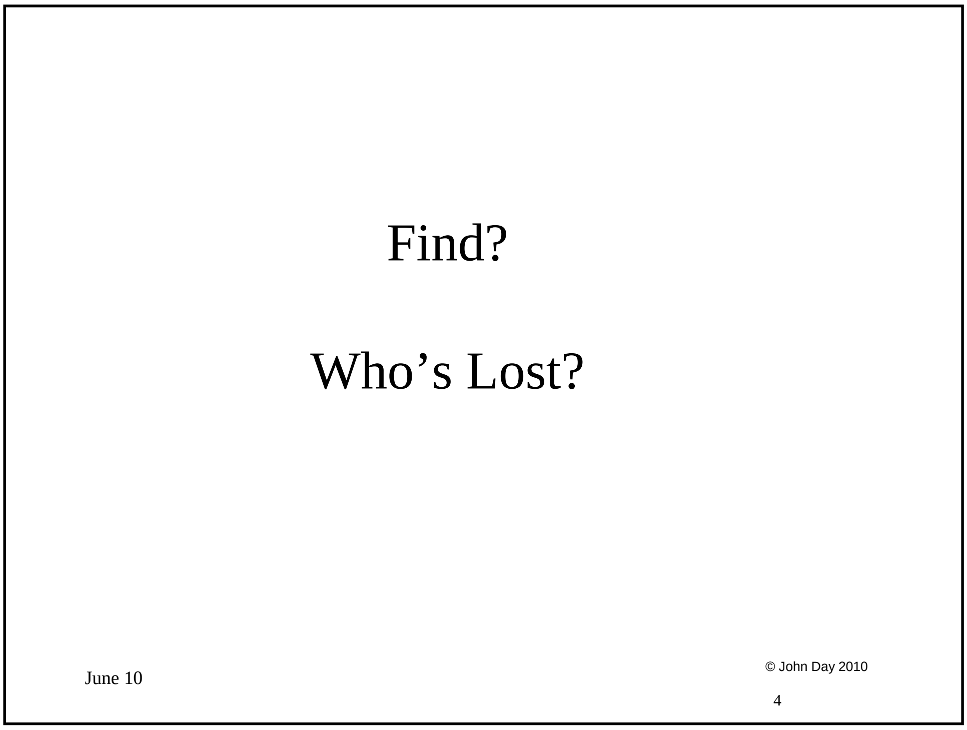


The Seven Unanswered Questions (cont)

- We needed application names and a directory.
  - What did they look like?
- We needed a solution to multihoming.
  - But being OS designers, the solution was clear.
  - But naming and addressing needed a hard look
- If addresses were location-dependent, what did that mean in a network?
    - This one is hard.
  - In essence, “we need an addressing architecture.”

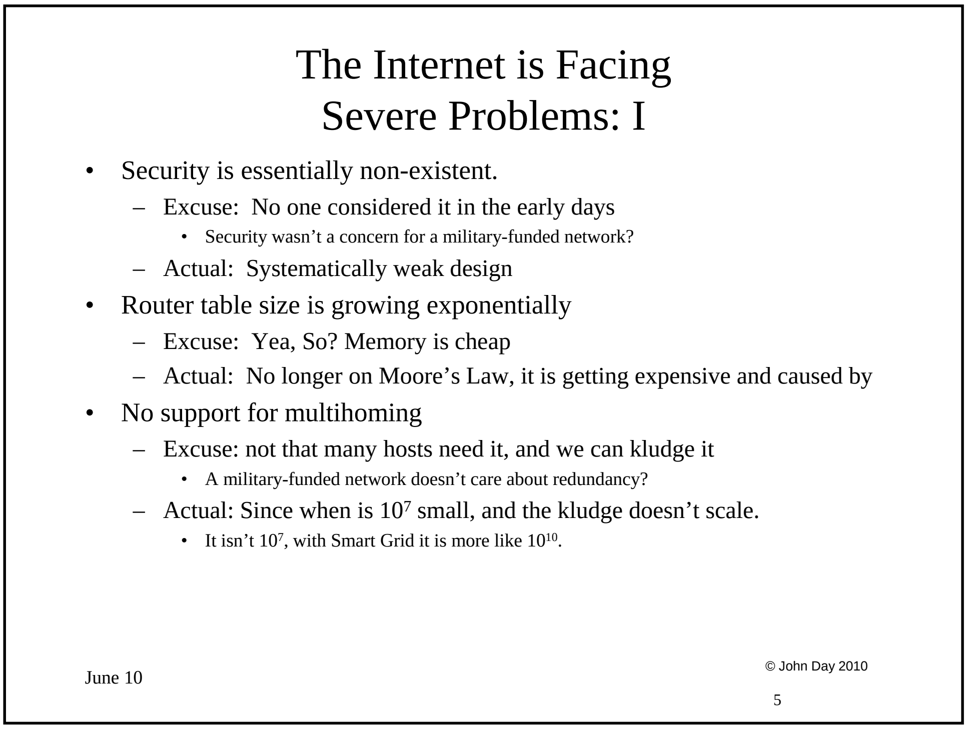
- Quite a List! Lots of material for researchers.
  - In the Internet these are still open issues.

See Preface of Patterns



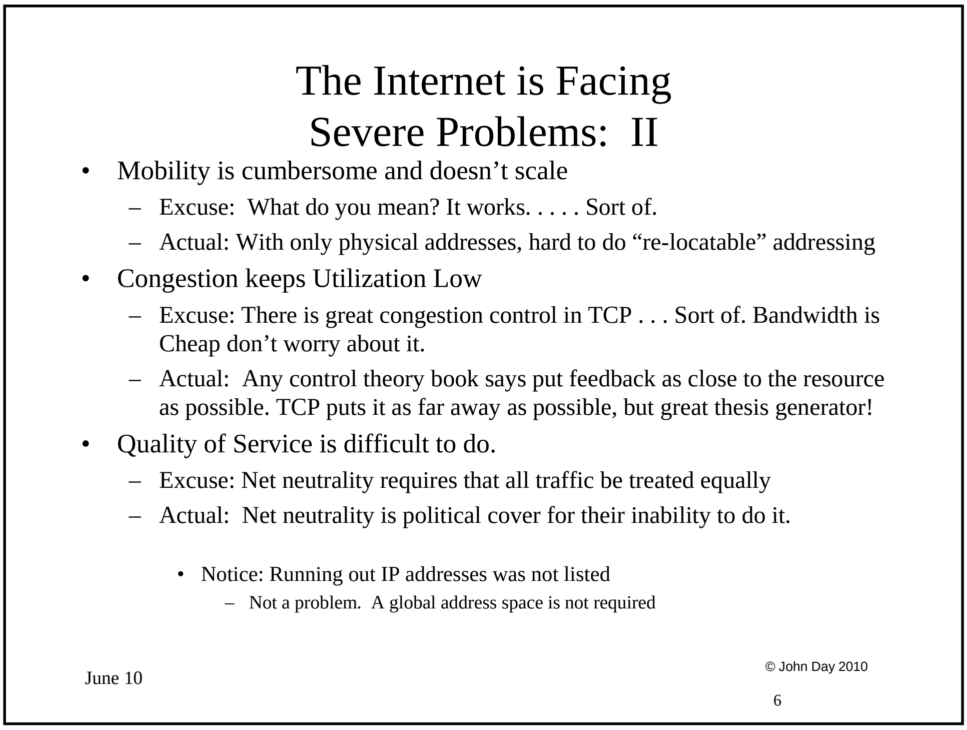
Find?

Who’s Lost?



The Internet is Facing Severe Problems: I

- Security is essentially non-existent.
  - Excuse: No one considered it in the early days
    - Security wasn't a concern for a military-funded network?
  - Actual: Systematically weak design
- Router table size is growing exponentially
  - Excuse: Yea, So? Memory is cheap
  - Actual: No longer on Moore's Law, it is getting expensive and caused by
- No support for multihoming
  - Excuse: not that many hosts need it, and we can kludge it
    - A military-funded network doesn't care about redundancy?
  - Actual: Since when is 10^7 small, and the kludge doesn't scale.
    - It isn't 10^7, with Smart Grid it is more like 10^{10}.



The Internet is Facing Severe Problems: II

- Mobility is cumbersome and doesn't scale
  - Excuse: What do you mean? It works. . . . . Sort of.
  - Actual: With only physical addresses, hard to do "re-locatable" addressing
- Congestion keeps Utilization Low
  - Excuse: There is great congestion control in TCP . . . Sort of. Bandwidth is Cheap don't worry about it.
  - Actual: Any control theory book says put feedback as close to the resource as possible. TCP puts it as far away as possible, but great thesis generator!
- Quality of Service is difficult to do.
  - Excuse: Net neutrality requires that all traffic be treated equally
  - Actual: Net neutrality is political cover for their inability to do it.

    - Notice: Running out IP addresses was not listed
      - Not a problem. A global address space is not required

And if that Weren’t Bad Enough

Much of What is Believed
about the Internet is Myth

© John Day 2010

Myths of the Internet: I

- The Internet is an Engine of Innovation.
  - The Internet has in a real sense been stagnant since the late-70s
  - Living on Moore’s Law and Band-aids
    - Lots of Innovation on *top* of the Internet, but even that has begun to wane.
- The Internet is decentralized. No one owns the Internet.
  - Actually, Same as the global PSTN, just no sexy name.
- The Internet is based on the ARPANET
  - Actually, It is based on CYCLADES
- The Internet is not an internet, but a catenet.
  - Ceased to be an Internet on January 1, 1983.
- The Internet is a dumb network.
  - Actually, it keeps maximal state in the network, not minimal.

© John Day 2010

Myths of the Internet: II

- The Internet has decentralized routing.
  - Actually, most ISPs routes are statically allocated.
- IP is the Internet Protocol.
  - That is what the letters stand for but it is really a subnet protocol.
- IP addresses name the host.
  - No, they name the interface to the host, same as a MAC address [sic]
- TCP isn’t perfect, but it is good enough.
  - Every design decision is not just wrong but makes something else worse. The one thing it got right was destroyed creating IP.

The Reality is Quite Different

- If the Internet were an Operating System, it would have more in common with DOS, than UNIX.
  - Been living on Moore’s Law and bandaids for 30 years.
- Imagine trying to use a modern computer with only physical memory addresses.
- That is the Internet today.

Need to Do Something About It

(Could start by just answering the 7 questions, . . . Huh!?)

- A Decade ago, DARPA funded NEWARCH
  - All the top minds to come up with a new Internet architecture
  - Two years later, they came up dry. Nada. Nothing.
- At this time, the National Research Council issued a report that said in part *“the insiders [network researchers] had not shown that they had managed to exercise the usual elements of a successful research program, so a back-to-basics message was fitting.”*
  - Must have been sobering.
- When DARPA was unwilling to throw good money after bad, they went to NSF to fund FIND and GENI, massive projects to FIND the Answer.
  - Been at it for years, spending millions, and have nothing to show.
  - An artisan’s response: when confronted with a problem, build something!
- Having not come up with anything, consensus is must look outside networking
  - A multidisciplinary approach!
  - Classic indicator of running out of ideas. Someone else must have them.

© John Day 2010

Guiding “Principles” Aren’t Much Help

- Soft State/Hard State
  - All properly designed protocols are soft state; only some database operations are hard state. A specious distinction
- Loc/Id Split
  - Post IPng trauma. Trying to avoid the obvious (and only) solution.
    - Real tip off: it’s a false distinction
  - Continues to route to the wrong place
- Fate Sharing
  - Mostly used as a rug . . . to sweep under.
- End-to-End Principle
  - At best a lemma. More a statement of desire, by focusing attention on the dichotomy of the middle vs the edge, it misses the point.
  - Hence becomes an impediment to finding a way forward.

© John Day 2010

The Myths and “Principles” are *the* Major Barrier to FINDing an Answer

- The field is no longer a science, but a craft.
  - They have been asking “What to build?”
  - Not asking, “What don’t we understand?”

- The Answer has been clear for a long time.

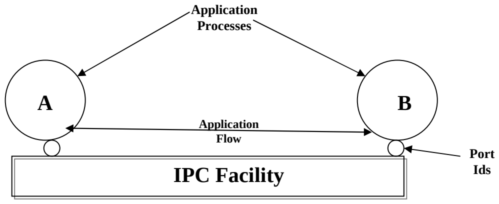
- And it is very simple:

- Networking is IPC and only IPC

© John Day 2010

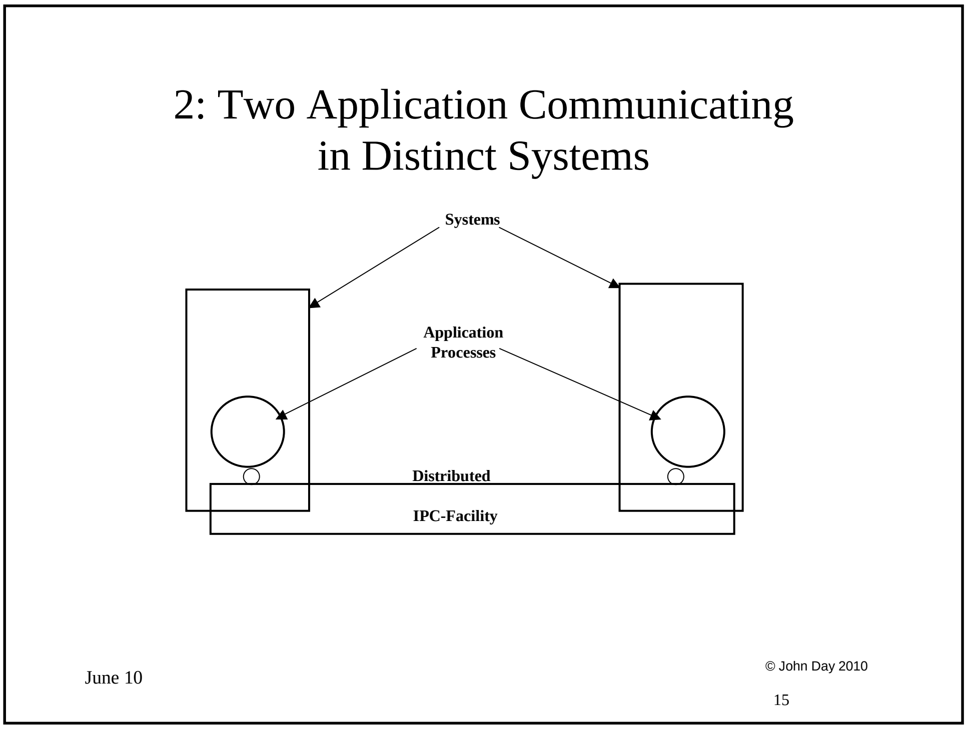
1: Start with the Basics

Two applications communicating in the same system.

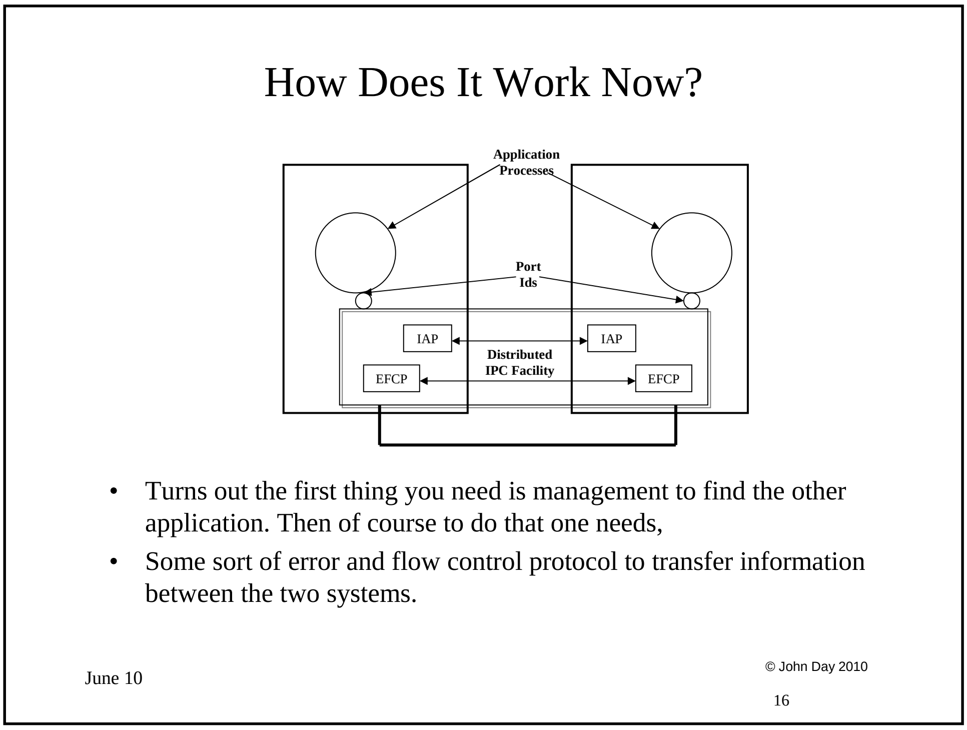


Communication within
a Single Processing System

© John Day 2010



2: Two Application Communicating in Distinct Systems



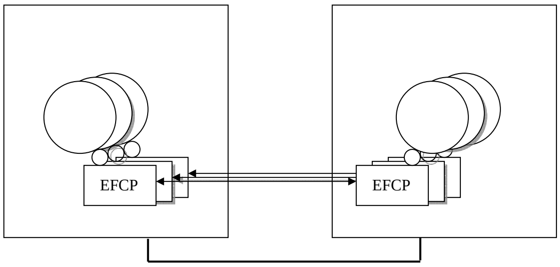
How Does It Work Now?

- Turns out the first thing you need is management to find the other application. Then of course to do that one needs,
- Some sort of error and flow control protocol to transfer information between the two 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.

Typically use the port-ids of the source and destination.

Connection-id

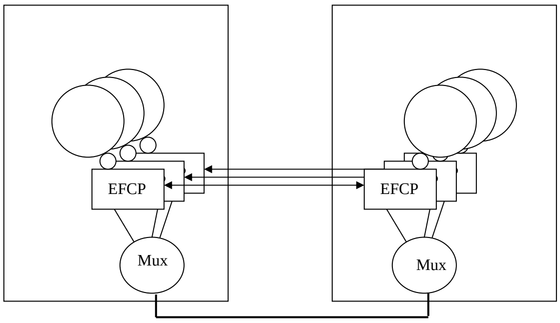
Dest-port	Src-port	Op	Seq #	CRC	Data

Also include the port-ids in the information sent in IAP to be used in EFCP synchronization (establishment).

Simultaneous Communication Between Two Systems

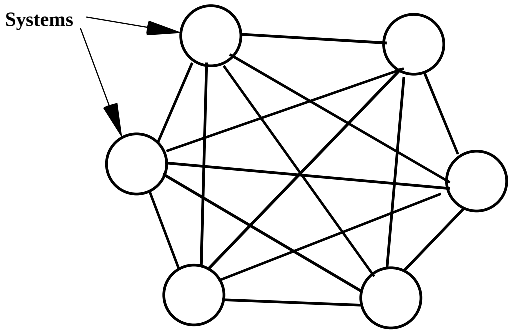
i.e. multiple applications at the same time

- Also Multiple Users of a Single Resource



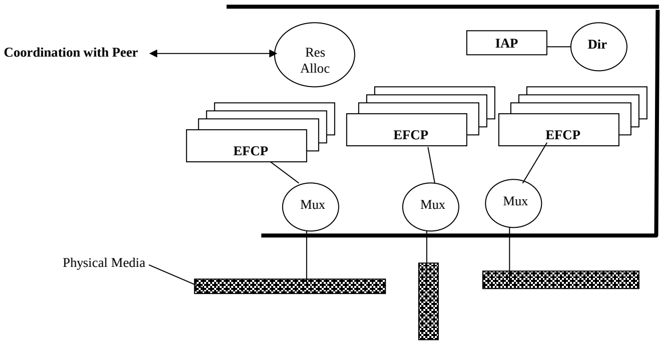
Will also need an application to manage multiple users of a single resource.

4: Communication with N Systems



© John Day 2010

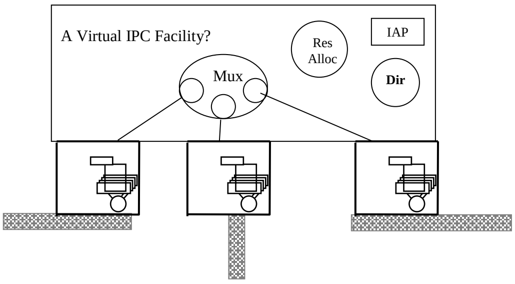
With Multiple Interfaces It Gets Messy



- So a task to manage the use of the interfaces and mask any differences.
  - A little organizing will help.

© John Day 2010

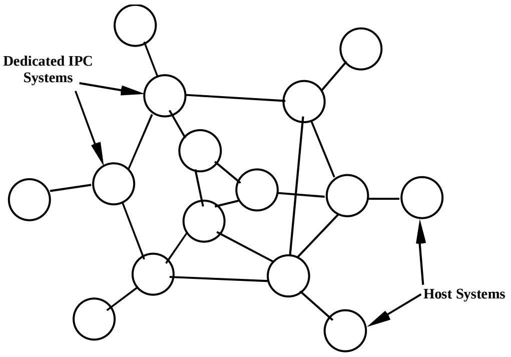
A Little Re-organizing



So we have a Distributed IPC Facility for each Interface and an application over all of them to manage their use and to give the user a common interface, a Virtual IPC Facility?

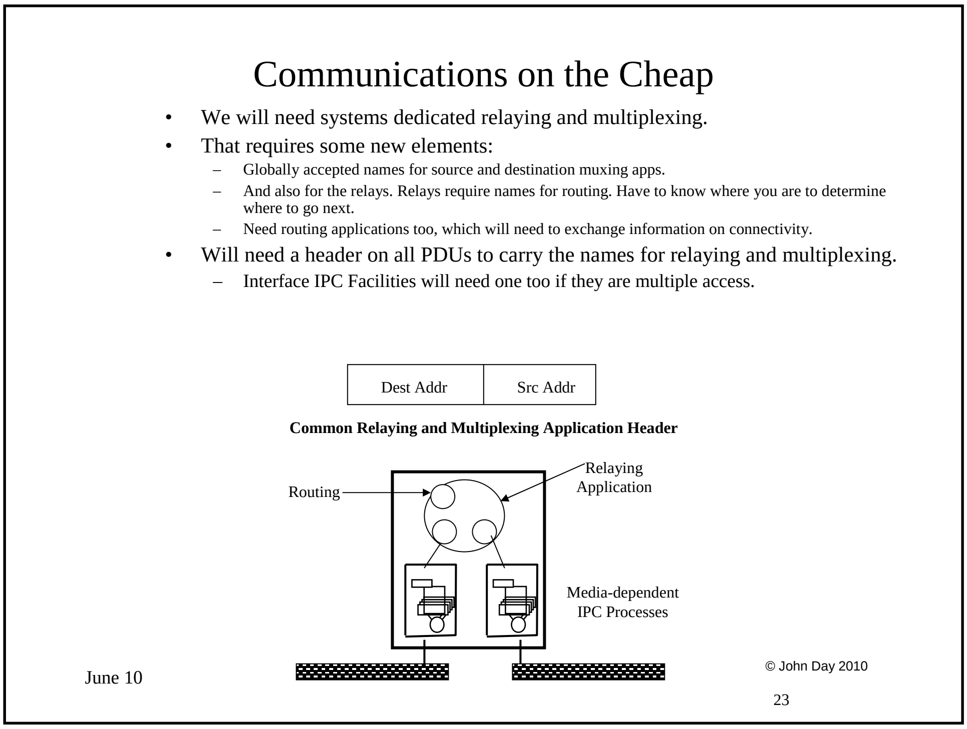
© John Day 2010

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.

© John Day 2010



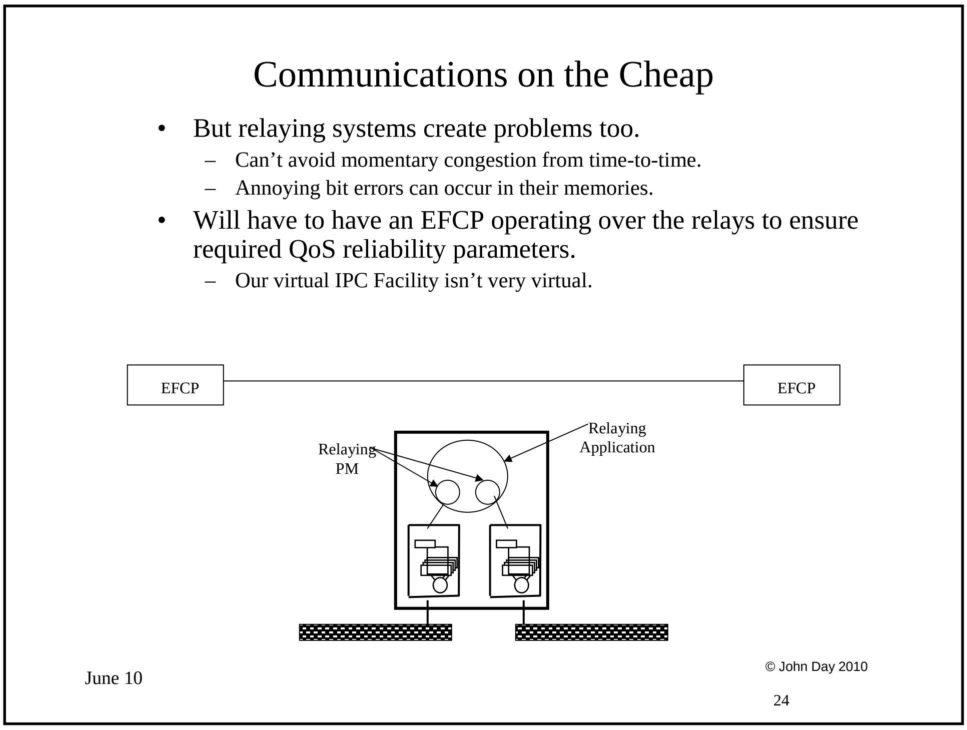
Communications on the Cheap

- We will need systems dedicated relaying and multiplexing.
- That requires some new elements:
  - Globally accepted names for source and destination muxing apps.
  - And also for the relays. Relays require names for routing. Have to know where you are to determine where to go next.
  - Need routing applications too, which will need to exchange information on connectivity.
- Will need a header on all PDUs to carry the names for relaying and multiplexing.
  - Interface IPC Facilities will need one too if they are multiple access.

Dest Addr	Src Addr

Common Relaying and Multiplexing Application Header

© John Day 2010

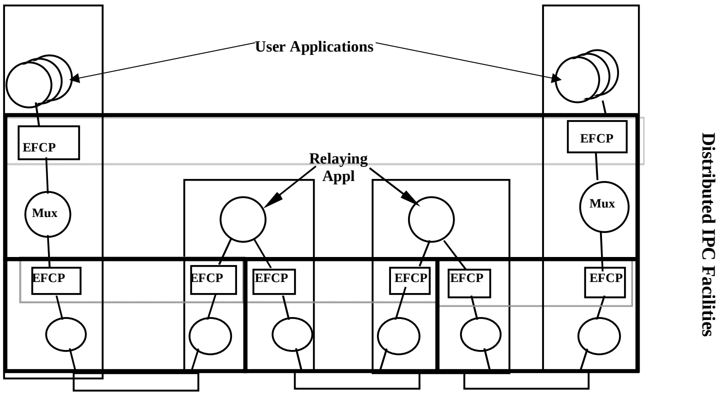


Communications on the Cheap

- But relaying systems create problems too.
  - Can’t avoid momentary congestion from time-to-time.
  - Annoying bit errors can occur in their memories.
- Will have to have an EFCP operating over the relays to ensure required QoS reliability parameters.
  - Our virtual IPC Facility isn’t very virtual.

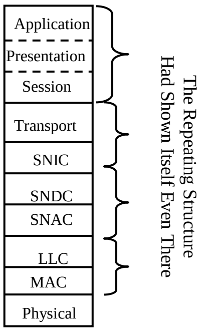
© John Day 2010

The IPC Model



© John Day 2010

We Should Have Seen It Sooner



© John Day 2010

The Implications

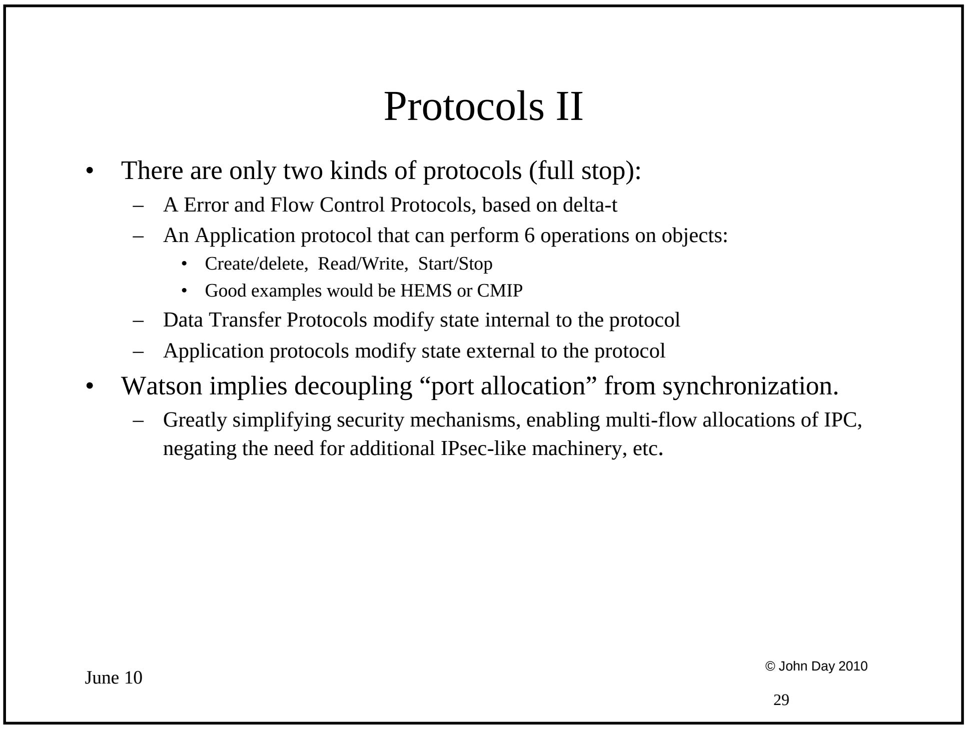
- Networking is IPC and only IPC.
  - We had been concentrating on the differences, rather than the similarities.
  - Of course, there are layers! What else do you call cooperating shared state that is treated as a black box? A waffle!?
- All layers have the same functions, with different scope and range.
  - Not all instances of layers may need all functions, but don’t need more.
- A Layer is a Distributed Application that performs and manages IPC.
  - A Distributed IPC Facility (DIF)
- This yields a theory and an architecture that scales indefinitely,
  - i.e. any bounds imposed are not a property of the architecture itself.
- Solutions to many “big problems” are a consequence of the structure.
- Addresses are internal to the layer. Application only knows destination application name and local handle.

© John Day 2010

Protocols I

- Looking at the Patterns in protocols there were other results that re-enforced the fundamentals.
- Watson proves that the *necessary and sufficient* conditions for distributed synchronization requires only that 3 timers are bounded:
    - Maximum Packet Lifetime
    - Maximum number of Retries
    - Maximum time before Ack
  - Yes, SYNs and FINs are superfluous
- This class of protocols naturally cleave into
  - data transfer - tightly bound, policy imposed by sender) and
  - Control - loosely bound, (feedback) policy imposed by receiver.
- Separating mechanism and policy in a delta-t like protocol will yield the entire range from UDP-like to TCP-like.

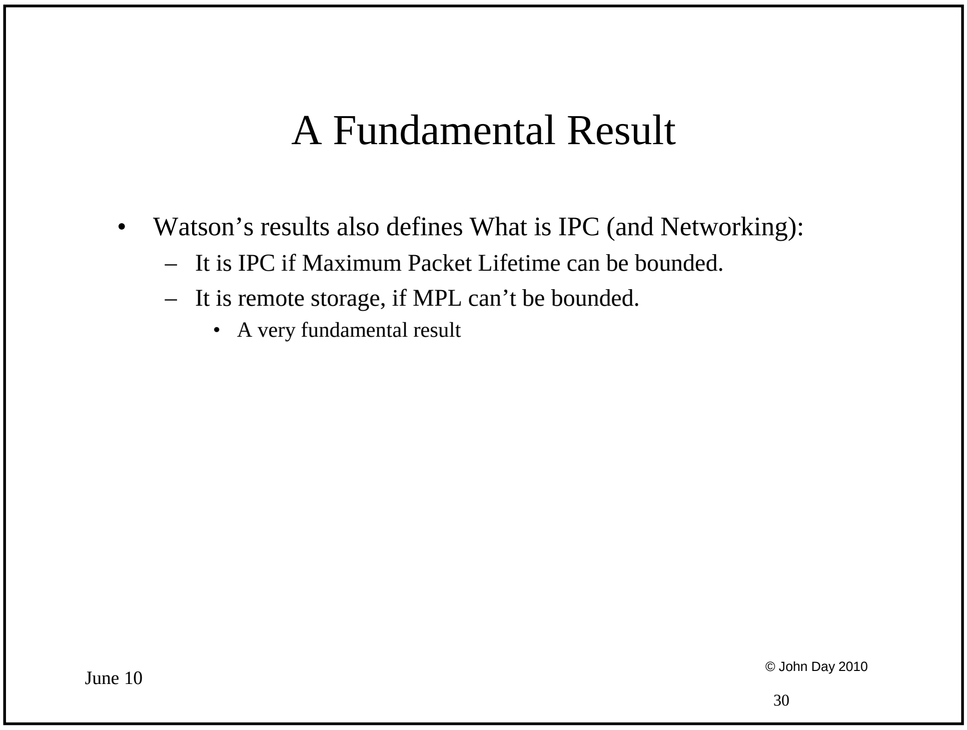
© John Day 2010



Protocols II

- There are only two kinds of protocols (full stop):
  - A Error and Flow Control Protocols, based on delta-t
  - An Application protocol that can perform 6 operations on objects:
    - Create/delete, Read/Write, Start/Stop
    - Good examples would be HEMS or CMIP
  - Data Transfer Protocols modify state internal to the protocol
  - Application protocols modify state external to the protocol
- Watson implies decoupling “port allocation” from synchronization.
  - Greatly simplifying security mechanisms, enabling multi-flow allocations of IPC, negating the need for additional IPsec-like machinery, etc.

© John Day 2010

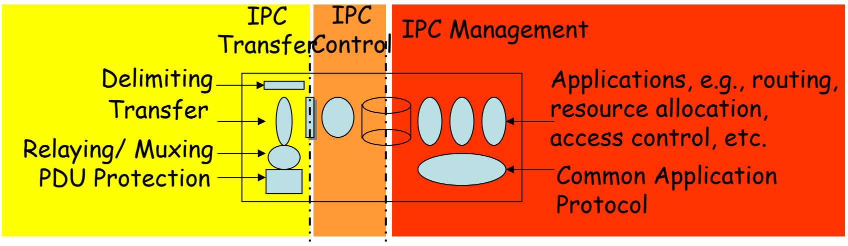


A Fundamental Result

- Watson’s results also defines What is IPC (and Networking):
  - It is IPC if Maximum Packet Lifetime can be bounded.
  - It is remote storage, if MPL can’t be bounded.
    - A very fundamental result

© John Day 2010

What a Layer Looks Like



IPC Transfer

IPC Control

IPC Management

Delimiting

Transfer

Relaying/ Muxing

PDU Protection

Applications, e.g., routing, resource allocation, access control, etc.

Common Application Protocol

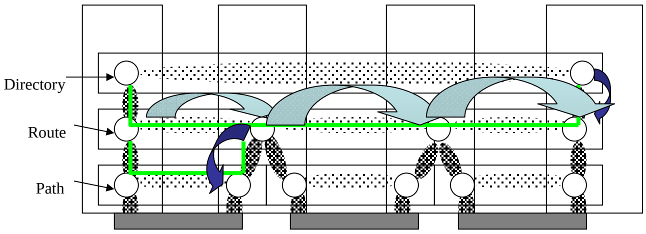
- A Layer consists of cooperating IPC Processes. Each one looks like this.
- Processing at 3 time-scales, decoupled by either a State Vector or a Resource Information Base
  – IPC Transfer actually moves the data (≈ IP + UDP)
  – IPC Control (optional) for retransmission (ack) and flow control, etc.
  – IPC Layer Management for routing, resource allocation, locating applications, access control, monitoring lower layer, etc.

Naming and Addressing I

- The IPC Processes are cooperating application processes dedicated to performing and managing IPC.
- Addresses are synonyms of the Application Process Names that are members of a layer.
  – Scope is limited to the layer and structured to facilitate use within the layer.
    - Can change an address without disturbing existing flows.
- Application Process Names (and their synonyms or sets of them) are the only non-local names required in an architecture.
  – All other names are local in scope.
    - Port-ids
    - Connection-endpoint-ids
- Saltzer [1982] worked out the naming and addressing architecture, but he missed a crucial case.
  – Networks are always more general than Operating Systems,

Generalizing Saltzer to Networks

- Directory maintains 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. (Saltzer missed this because they hadn’t occurred yet.)
- 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.

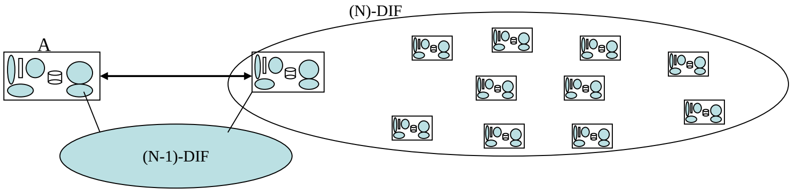


Naming and Addressing II

- A Layer routes on the addresses in its layer.
  - (Sounds obvious but IP routes on someone else’s address)
- Routing table size can be bounded (implication of recursion).
- Deep packet Inspection is unnecessary.
- Multihoming provided at no cost, a consequence of the structure
  - Old result (1982): Don’t embed an (N-1)-address in an (N)-address.
    - Makes it a *pathname* and defeats necessary properties of path independence
- Mobility is just dynamic multihoming with expected failures.
  - Only difference is the frequency of changes.
- Multicast and anycast are degenerate cases of whatevercast:
  - A whatevercast name is the name of a set of addresses with a rule for selecting members of the set when the whatevercast name is referenced.
  - Multicast folds into unicast and vice versa.

How Does It Work?

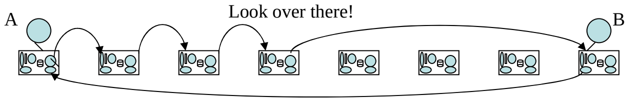
Joining or Creating a Layer



- Nothing more than Applications establishing communication (for management)
  - Authenticating that A is a valid member of the (N)-DIF
  - Initializing it with the current information on the DIF
  - Assigning it a synonym to facilitate finding IPC Processes in the DIF, i.e. an address

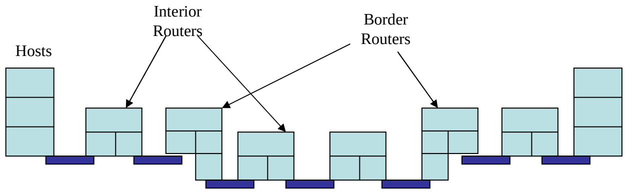
How Does It Work?

Establishing Communication



- Simple: do what IPC tells us to do.
  - A asks IPC to allocate comm resources to B
  - Determine that B is not local to A use search rules to find B
  - Keep looking until we find an entry for it.
  - Then go see if it is really there and whether we have access.
  - Then tell A the result.
- This has multiple advantages.
  - We know it is really there.
  - We can enforce access control
  - We can return B’s policy and port-id choices
  - If B’s has moved, we find out and keep searching

Only Three Kinds of Systems

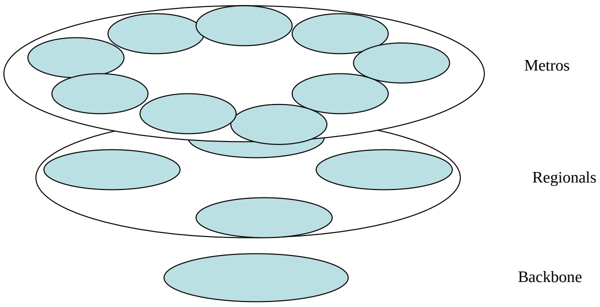


- Middleboxes? We don’t need no stinking middleboxes!
- NATs: either no where or everywhere,
  - NATs only break broken architectures
- The *Architecture* may have more layers, but no *box* need have more than the usual complement.
  - Hosts may have more layers, depending on what they do.

How Does It Work?

The Internet and ISPs

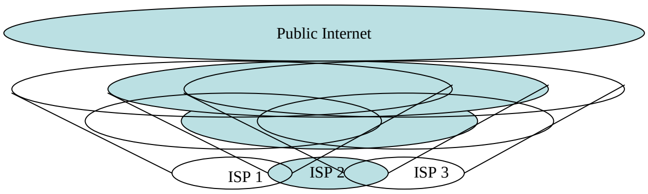
- ISPs have as many layers as they need to best manage their resources.



How Does It Work?

The Internet and ISPs

- The Internet floats on top of ISPs, a “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.

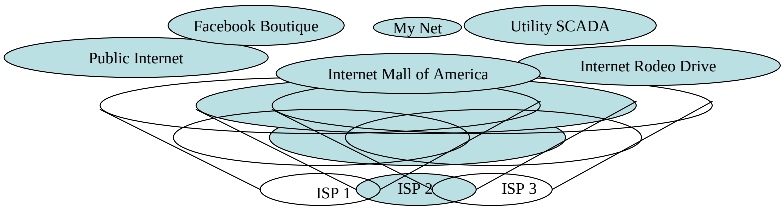


© John Day 2010

How Does It Work?

The Internet and ISPs

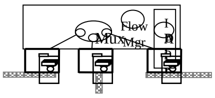
- But there does not need to be ONE e-mall.
  - You mean!
    - Yes, it is really an INTERnet!



© John Day 2010

Choosing a Layer

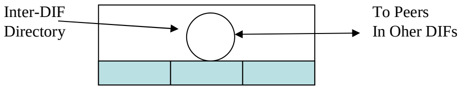
- In building the IPC Model, the first time multiple DIFs are encountered (data link layers in that case), it was found useful to have a task to determine which DIF to use.



  - So user didn’t have to see all of the wires
  - But the user shouldn’t have to see all of the “Nets” either.

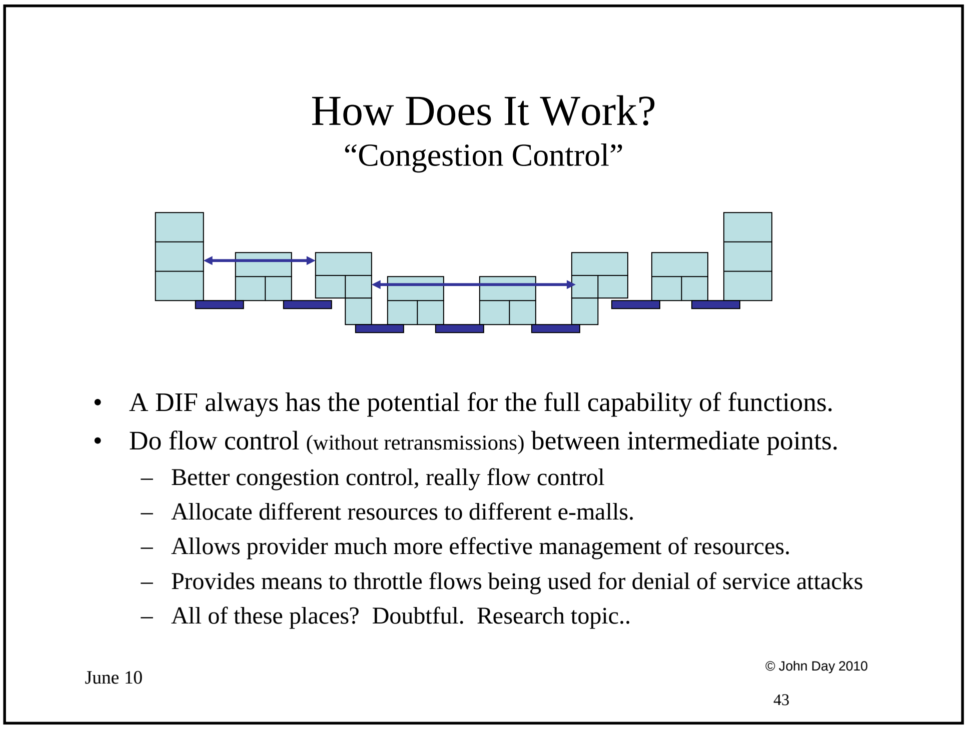
- This not only generalizes but has major implications.
- A Global Address Space is unnecessary.

A Global Application Name Space is Useful, but a Global Address Space is Not Necessary



Actually one could still have distinct names spaces within a DIFs (synonyms) with its own directory database.

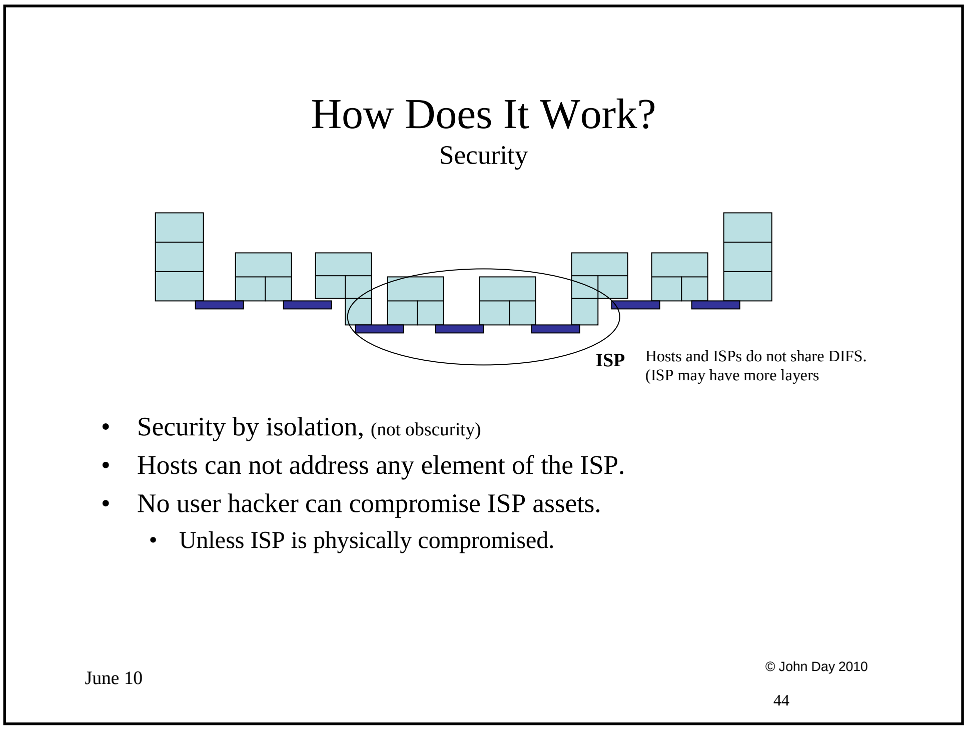
- This is how networks get generated.
- But not all names need be in a Global Directory.
- A DIF could have its own parallel application name space and directory of distributed databases.
- This means that alternate application name spaces and directory schemes are not only possible, but useful.



How Does It Work?

“Congestion Control”

- A DIF always has the potential for the full capability of functions.
- Do flow control (without retransmissions) between intermediate points.
  - Better congestion control, really flow control
  - Allocate different resources to different e-malls.
  - Allows provider much more effective management of resources.
  - Provides means to throttle flows being used for denial of service attacks
  - All of these places? Doubtful. Research topic..



How Does It Work?

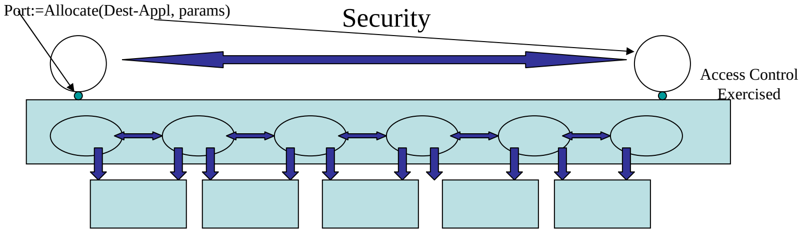
Security

Hosts and ISPs do not share DIFS.
(ISP may have more layers

- Security by isolation, (not obscurity)
- Hosts can not address any element of the ISP.
- No user hacker can compromise ISP assets.
  - Unless ISP is physically compromised.

How Does It Work?

Security



- The DIF is a securable container. DIF is secured not each component separately.
- 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.

June 10

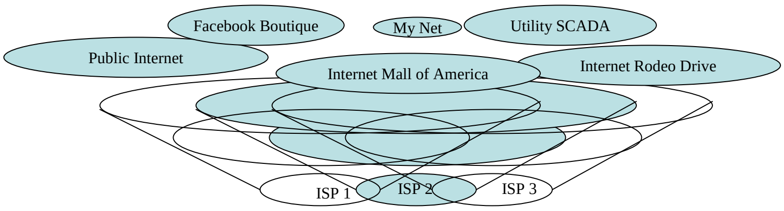
© John Day 2010

45

How Does It Work?

Security

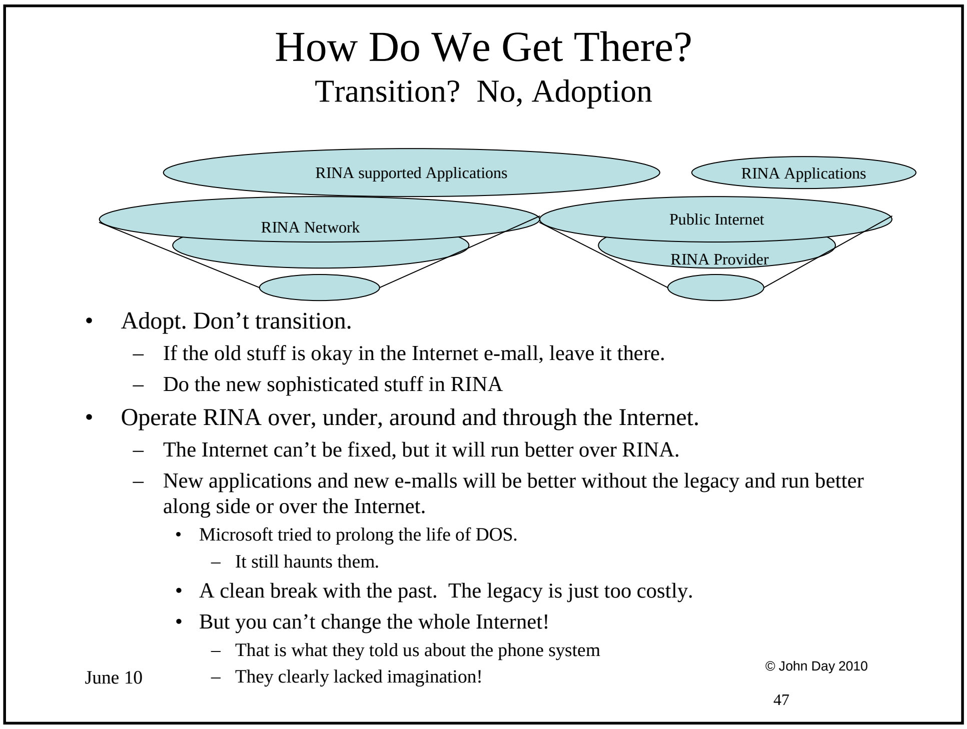
- A Hacker in the Public Internet cannot connect to an Application in another DIF without either joining the DIF, or creating a new DIF spanning both. Either requires authentication and access control.
  - Non-IPC applications that can access two DIFs are a potential security problem.



June 10

© John Day 2010

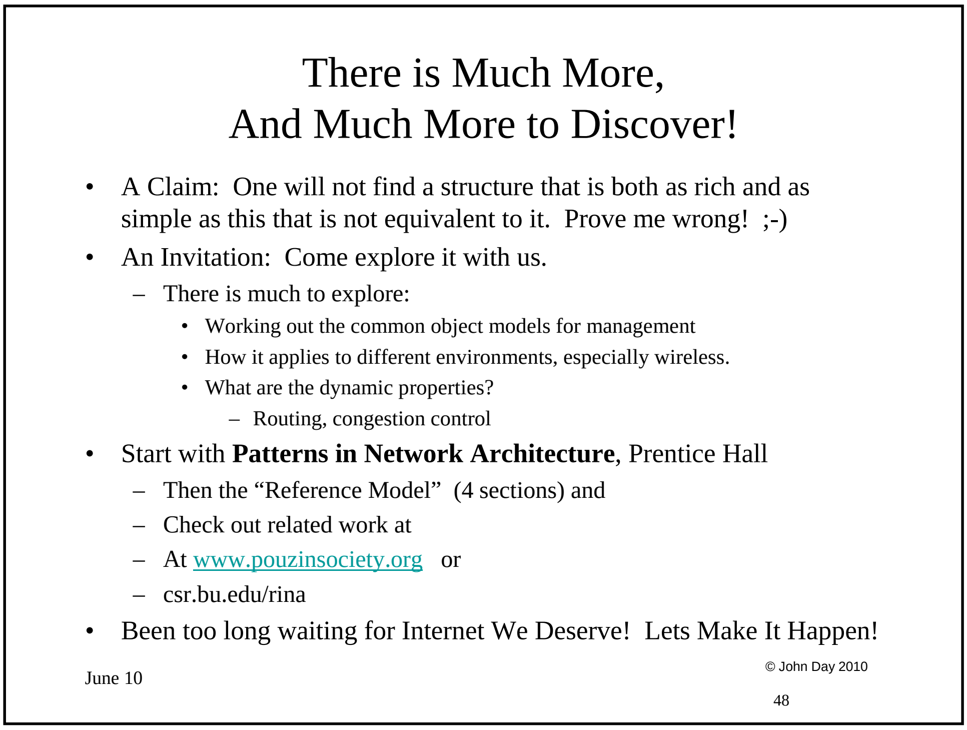
46



How Do We Get There?

Transition? No, Adoption

- Adopt. Don’t transition.
  - If the old stuff is okay in the Internet e-mall, leave it there.
  - Do the new sophisticated stuff 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.
    - Microsoft tried to prolong the life of DOS.
      - It still haunts them.
    - A clean break with the past. The legacy is just too costly.
    - But you can’t change the whole Internet!
      - That is what they told us about the phone system
      - They clearly lacked imagination!



There is Much More, And Much More to Discover!

- A Claim: One will not find a structure that is both as rich and as simple as this that is not equivalent to it. Prove me wrong! ;-)
- An Invitation: Come explore it with us.
  - There is much to explore:
    - Working out the common object models for management
    - 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
  - csr.bu.edu/rina
- Been too long waiting for Internet We Deserve! Lets Make It Happen!