# Architectures for an Internet

David Clark

MIT CSAIL

October 2021

# Preface: the term *architecture*

We borrowed the term…

A process, an outcome, a discipline.

- Putting components and design elements together to make an entity that serves a purpose.
  - Modularity, interfaces, dependency, layering, abstraction and component reuse.

How do “real” architects learn their trade? Case studies.

- Architecture is a design discipline.
  - *Design patterns – Christopher Alexander*

# Elements of network architecture

- Issues on which we must agree for the system to function.
  - Internet: common packet header, global address space (wrong), a common global routing scheme, no flow setup.
- Issues on which it is convenient to agree.
  - The Domain Name System
- Basic modularity of the system.
  - IP sits above comms technology and below applications.
    - Interface: a constraint that deconstrains. (John Doyle)
  - There are autonomous regions of the network.
- Functional dependencies and service models.
- Durable assumptions.
  - How does a system evolve?
- All in the context of “what is its purpose”?

# Design considerations

- Fit for purpose
  - Think hard about requirements.
    - Example: generality vs. focus on key application.
  - The functional specification of the Internet is somewhat implicit.
    - A consequence of the fact we were exploring the unknown. Not so today.
- Allocation of function
  - Which part should do what?
  - Layered systems present a specific version of this question.
  - If the lower layer does X, done once, done by experts, better quality?
    - Becomes a service, but with rigid functionality.
  - If the upper layer does X, more adaptable.
- OS as example.

# Functional spec of the Internet

(Somewhat implicit, as I said.)

A packet with a valid destination address will be delivered to those destinations with best effort.

- Allowed to fail to an unspecified degree.
- What comes out is what went in (simple version).
- What comes out is semantically equivalent to what went in (disputed).
- No specification of what should *not* happen.
  - Good idea—do not impose requirements the architecture cannot enforce.
- No specification of how network resources are shared.
  - In a shared system, a normal requirement for the infrastructure is that it isolate and protect users.

# Placement of function

- A CS view: balance of convenience vs. flexibility.
  - If a single design meets a broad range of needs, why not make it a service?
- A real-world view: modularity induces industry structure.
  - See Ronald Coase: industry boundaries occur where the interfaces are clear.
    - The artful removal of interfaces…
  - We have firms that play the role of ISPs because the ISP function is specified by the IP specification, the global routing specification, etc.
  - A different functional placement would give the ISPs a different role to play, e.g., a different business.
- Examples: DNS, identity.

# Real world considerations

- If architecture induces industry structure, it allocates control of function to different actors.
- If it allocates control, it allocates power.
- In a multi-party ecosystem like the Internet:
  - **ARCHITECTURE IS POLITICS.**
- We don’t teach this in CS classes, but you better understand it.
  - Read about political economy.
- Lots of actors want power and control.
  - Private firms: thrive, make money, grow.
  - Sovereign states: control content, surveillance, taxation, etc.

# Isolating users

- On the Internet, anyone can send packets at will.
  - No flow setup, etc.
- ISPs throttle users for business reasons.
- But no commitment to fairness, regulation of abuse, etc.
  - Fair on what basis?
  - Flows (the Internet does not manage them).
  - Packet counts? Over what period?
- Today, regulation of sending is assigned to the end-nodes, not the network.
  - But why trust the end-nodes?
- Is this a fundamental failure of layered functionality? See:
  - Lloyd Brown, Ganesh Ananthanarayanan, Ethan Katz-Bassett, Arvind Krishnamurthy, Sylvia Ratnasamy, Michael Schapira, and Scott Shenker. 2020. On the Future of Congestion Control for the Public Internet. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks* (*HotNets '20*). Association for Computing Machinery, New York, NY, USA, 30–37. DOI:https://doi.org/10.1145/3422604.3425939

# The functional dispute

- In the Internet, moving function “up” is either:
  - Adding higher-layer function to the core network.
  - Moving function “out” to the edge-devices.
- The end-to-end argument is an argument to move function up and **out**.
  - Putting more function “in” the network:
    - Will limit generality and limit the range of apps the net can support.
    - Will NOT be successful because the edge cannot trust the network to implement the function correctly.
      - End node will have to verify that the function was correctly performed, so will end up owning the problem anyway.
    - Success of packet delivery is self-evident. Beyond that it can get tricky.
    - Saltzer, J. H., Reed, D. P., & Clark, D. D. (1984). End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)*, *2*(4), 277-288.

# More examples of adding function to the net:

- Quality of service (service allocation).
  - Failed in the Internet for economic reasons. Widely used in specific cases.
    - Claffy, K. C., & Clark, D. D. (2016). Adding enhanced services to the internet: Lessons from history. *Journal of Information Policy*, *6*(1), 206-251.
- Multicast.
  - Failed in general (economics again), but heavily used in specific cases.
- Anycast.
  - Not really a change, just an agreement about acceptable use.
  - In wide use today.
- Source routing.
  - In the initial design—operational failure.

# The centrality of trust

- By allocating a function to an actor, either:
  - There are constraints wrapped around that actor/function to verify correct operation.
  - We place trust in the actor to function as specified.
- A lot of the Internet runs on trust.
  - DNS gives the right answer.
  - Certificate Authorities are trustworthy.
- A design principle: trust but verify.
  - Example: how could we give ISPs the function of protecting confidentiality of content?

# On to specific architectural proposals

- Organized around particular functional objectives

# Abstracting the architectural invariants

- Serpent (1989)
  - Source route, global name space (tricky),  User control over routing.
- Application Layer Framing (ALF) (1990)
  - Application Data Units with conversion.
  - Also had performance objective. Reduce copy overhead in end-point.
- Metanet (1997)
  - “We argue that a new architectural component, the region, should form a central building block of the next generation network.”
- Plutarch (2003)
  - How minimal can you be? Source routes, conversion boxes, gossip
- Framework for Internet Innovation (FII) (2011)
  - Assumes specific architectures created in this framework.  Abstract service semantics.

# Diverse network technology

- An old problem—it motivated the Internet.
  - Original motivation: hook together ARPAnet, Satnet, PacketRadio network.
- These networks offered different delivery semantics.
  - Cannot easily convert between them.
- Solution: a “spanning” or “overlay” architecture that defines a new service model and uses the features of the underlay networks to build it.
- Alternative: “conversion” approach where the abstract service model is more similar, and an interconnection point maps between them.
  - Metanet, Serpent, Plutarch, ALF, FII
- Today the Internet “spans” Ethernet, MPLS, etc.
  - The simplicity of the Internet service model makes this easier.

# Further reading


- Cheriton, David R. "Sirpent: A high-performance internetworking approach." *Symposium proceedings on Communications architectures & protocols*. 1989.
- Clark, D. D., & Tennenhouse, D. L. (1990). Architectural considerations for a new generation of protocols. *ACM SIGCOMM Computer Communication Review*, *20*(4), 200-208.
- Wroclawski, John. "The metanet: White paper." *Workshop on Research Directions for the Next Generation Internet*. 1997.
- Crowcroft, Jon, et al. "Plutarch: an argument for network pluralism." *ACM SIGCOMM Computer Communication Review* 33.4 (2003): 258-266.
- Koponen, T., Shenker, S., Balakrishnan, H., Feamster, N., Ganichev, I., Ghodsi, A., ... & Kuptsov, D. (2011). Architecting for innovation. *ACM SIGCOMM Computer Communication Review*, *41*(3), 24-36.

# Higher-level names in the architecture

“Information-centric networking”

- Triad (2000)
  - URL in packet header. Sets up traditional flow using addresses.
- Dona (Data-Oriented Network Architecture) (2007)
  - Flat names.
- PSIRP/PURSUIT (2008,2012) Publish-subscribe conception. Names are nested scopes.
- Netinf (2013)
  - Flat names like DONA. Supports regional diversity. Discusses options for name management.
- NDN (2014)
  - No routed addresses. All packets contain “names”. Per packet state in routers.

# Named Data Networking

- All packets routed using a DNS-like name.
- An end point that wants a packet of data sends an “interest” packet.
  - This is routed toward the data.
  - Routers remember where the *interest* packet came from.
  - Implies that the requestor knows the name.
- When the *interest* packet encounters the desired data, a *data* packet is returned.
  - Stored state in the routers direct it toward the sender of the *interest.*
- Packets have no source address.
- Routers can cache data for efficiency.

# Major concerns

- How can routing scale if the names do not express topology?
  - Hierarchical, flat…
- Who should control the name-location binding?
  - The network?
  - The owner of the name?
- Ask: what concerns should influence the binding?
  - Hint: not just technical (e.g., performance).

# For further reading


- Cheriton, David R. *TRIAD: Translating Relaying Internetwork Architecture Integrating Active Directories*. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 2003.
- Koponen, T., Chawla, M., Chun, B. G., Ermolinskiy, A., Kim, K. H., Shenker, S., & Stoica, I. (2007, August). A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications* (pp. 181-192). (DONA)
- Fotiou, N., Nikander, P., Trossen, D., & Polyzos, G. C. (2010, October). Developing information networking further: From PSIRP to PURSUIT. In *International Conference on Broadband Communications, Networks and Systems* (pp. 1-13). Springer, Berlin, Heidelberg.
- Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K. C., Crowley, P., ... & Zhang, B. (2014). Named data networking. *ACM SIGCOMM Computer Communication Review*, *44*(3), 66-73.
- Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B., & Karl, H. (2013). Network of information (netinf)–an information-centric networking architecture. *Computer Communications*, *36*(7), 721-735.

# Supporting evolution

- Two approaches.
  - Define fixed, abstract service model. Replace technology under that service model.
  - Imbed paths for evolution within a more concrete architecture.
- eXtensible Internet Architecture (XIA)(2011)
  - Multiple “kinds” of names in destination address. Content, service, host.
    - Organized as a DAG.
    - Different kinds of names can trigger different forwarding semantics.


- Anand, A., Dogar, F., Han, D., Li, B., Lim, H., Machado, M., ... & Steenkiste, P. (2011, November). XIA: An architecture for an evolvable and trustworthy Internet. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (pp. 1-6).

# Intermittent connectivity

- Delay/Disruption tolerant network (DTN).
  - Relay of ADUs (bundle)
  - Extreme regional variation. (Interplanetary links, data mules).
  - Requires a new service model: Reliable store and forward.
  - Addresses of the form <region:entity>
    - Globally mapped region topology. (Like BGP)
  - Example of a spanning architecture.
- The layering question again—which layer does what?


- Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., ... & Weiss, H. (2003). Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, *41*(6), 128-136.

# Shaping industry structure

- Some history: in 1992 then Senator Gore announced his vision for the National Information Infrastructure (NII).
  - Industry took note.
- Computer Systems Policy Project (CSPP, 1994)
  - A requirements document.
    - Computer Systems Policy Project. 1994. Perspectives on the National Information Infrastructure: Ensuring Interoperability.
- Cross-Industry Working Team (XIWT, 1994)
  - More of an architectural specification.
  - Defined two “new” interfaces: Network/Network, Network Service Control Point
    - Cross Industry Working Team. (1994). An Architectural Framework for the National Information Infrastructure. *White paper, September*.

# Mobility

The problem: when a end-point moves it changes its location but not its identity.

- Internet muddles the two. (Used IP address in TCP pseudo-header.)
- Must disentangle the two concepts. (Security concerns follow…)
- Devise a scheme to deal with rapid changes in location

MobilityFirst (MF, 2012)

- Naming:  DNS to GUID (flat), GNS maps GUID to <NA/entity>
- Both in header. Router look up new NA/entity if entity has moved.
- More complicated service model: caching.
- Key design question: is GNS global or end-point controlled?
  - Raychaudhuri, D., Nagaraja, K., & Venkataramani, A. (2012). Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, *16*(3), 2-13.
  - Arun Venkataramani (Umass) has papers on how to build a GNS.

# Avoiding global identifiers

Three issues with global identifiers:

- How to issue them; how to route on them , who can see them (security issue).
- Using a hash of content gives ”pretty unique” identifiers with security benefits.
  - But big: SHA-3 is 28-64 bytes. You want that in every header?

- MobilityFirst: <NA:entity> but also GUID.

Idea: Unique identifiers should be a private matter among end points. Not visible in the network.

- Newarch project (2004)
  - Association and Rendezvous Architecture (FARA)
  - Forwarding Directive, (source route). Rendezvous server (tricky). Used for mobility.
    - Clark, David, Karen Sollins, John Wroclawski, Dina Katabi, Joanna Kulik, Xiaowei Yang, RobertBraden, Aaron Falk, Venkata Pingali, Mark Handley, and Noel Chiappa. 2004. New Arch: Future Generation Internet Architecture. Available at http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf

# Conceptual clarity

- Recursive Network Architecture (RNA, 2006) and Recursive InterNetwork Architecture (RINA, 2008)
- Basic idea: different layers in the network system have structural similarity. Embed this idea in the architecture.
  - Network (like MPLS, ARPAnet). Internetworking (like IP). Why only two?
  - A layer forms an association among its endpoints to create an abstraction.
  - Distinct addressing matched to the need/**scope** of the layer.
    - IP reused in unclean way.
- Key architectural element: flow of info to support cross-layer functions.
  - Why we get “layer 2.5” devices in the Internet. Messy.
- Debate: who controls multi-homing?
- Concept or code?

# For further reading

- Touch, J., Wang, Y. S., & Pingali, V. (2006). A recursive network architecture. *ISI, Tech. Rep*, *626*.
- For RINA, I suggest starting with the Wikipedia page.

# Service composition

- Internet Indirection Infrastructure (I3, 2002)
  - Servers that receive packets sent to an identifier and forward them to intended receiver.
  - Receiver initiates the identifiers. Senders cannot send at will.
  - Both sender and receiver can cascade servers, which can do more than forward (term: per hop behavior or PHB).
  - Scale issue, which server hosts the identifier. Solution, DHT.
- Delegation Oriented Architecture (DOA, 2004)
  - More efficient forwarding.
- Nebula (2014)
  - NVENT control plane takes service requirements from sender and receiver and builds a source route (Proof of Consent), which is transformed into a Proof of Path. See also the ICING paper.
- Scion (2011)
  - Similar to Nebula. Strong emphasis on security.
- ChoiceNet (2014)
  - Emphasizes the economic framework around service composition. Not a data plane architecture.

# Friend or foe?

- Systems like i3 can be used in two modes:
  - The end-nodes trust each other.
  - The end-nodes do not.
- In the former case, the service elements are probably functional.
- In the latter case, they will probably play a role related to security.
  - Firewall, content checking, DDoS prevention, etc.
- Design challenge: can you build a system that is equally fit for both roles?

# For further reading


- Stoica, I., Adkins, D., Zhuang, S., Shenker, S., & Surana, S. (2002, August). Internet indirection infrastructure. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (pp. 73-86).
- Naous, J., Walfish, M., Nicolosi, A., Mazieres, D., Miller, M., & Seehra, A. (2011, December). Verifying and enforcing network paths with ICING. In *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies* (pp. 1-12).
- Anderson, T., Birman, K., Broberg, R., Caesar, M., Comer, D., Cotton, C., ... & Yoo, C. S. (2014). A brief overview of the NEBULA future internet architecture. *ACM SIGCOMM Computer Communication Review*, *44*(3), 81-86.
- Walfish, M., Stribling, J., Krohn, M. N., Balakrishnan, H., Morris, R. T., & Shenker, S. (2004, December). Middleboxes No Longer Considered Harmful. In *OSDI* (Vol. 4, pp. 15-15). (DOA)
- Zhang, X., Hsiao, H. C., Hasker, G., Chan, H., Perrig, A., & Andersen, D. G. (2011, May). SCION: Scalability, control, and isolation on next-generation networks. In *2011 IEEE Symposium on Security and Privacy* (pp. 212-227). IEEE.
- Barrera, D., Chuat, L., Perrig, A., Reischuk, R. M., & Szalachowski, P. (2017). The scion internet architecture. *Communications of the ACM*, *60*(6), 56-65.

# Active networking

A different idea: let the packet carry code that the router executes to express the treatment of the packet. (1996)

- The router has functions (PHBs). Compose them.
- ANTS (1999)
  - Packet carries the hash of the code. Who gets to install?
- Service composition and glue code (Smart Packet (1999), PAN (1999) PLANet.
- Challenges: performance, size of code, security (isolation).
- Objectives: controlling performance; diagnosis

# For further reading


- Tennenhouse, D. (1996). Active networks. In *{USENIX} 2nd Symposium on {OS} Design and Implementation ({OSDI} 96)*.
- Wetherall, D. J., Guttag, J. V., & Tennenhouse, D. L. (1998, April). ANTS: A toolkit for building and dynamically deploying network protocols. In *1998 IEEE Open Architectures and Network Programming* (pp. 117-129). IEEE.

# General thoughts

- Identity, naming addressing, location.
  - How are they split up, how are they mapped? Who can see what?
- Service model
  - Composable building blocks.
  - A few service classes (XIA and MF).
  - Single new service (NDN)
- Minimality vs. expressive power.
- Economics: the role of the ISP.
  - NDN-caching.
- Adverse interests.

# Classes of requirements

- Fit for purpose
  - Define the purpose
- Longevity
- Security
- Availability
- Economics
  - Are all the entities induced by the architecture viable economically?
- Management
- Meet the needs of society.