slices RI

PAWR Project Office

acm sigcomm

theNetworking Channel

Lessons learned from 40+ years of the Internet

10/4/2023

https://networkingchannel.eu

Panelists

theNetworking Channel

Henning Schulzrinne
Columbia U.

David Oran
MIT

Jörg Ott
TU München

Craig Partridge
Colorado State U

John Wroclawski
Information Sciences Institute

Organizers

H. Schulzrinne
Columbia U

J. Kurose
U Massachusetts

Lessons learned from 40+ years of the Internet

Dagstuhl seminar: May 2023 https://www.dagstuhl.de/23182

Reflection: (40+ years experience) general design principles for network protocols, algorithms:

- Which approaches have aged well? Why?
- Not just data transport, also control and management
- “Do-over”: what considerations, principles would guide the design?
- “Beyond the technical”: implementation strategies, deployment incentives, economics, regulation affect protocols and their evolution as well
- Case studies of successful protocol developments/deployments as well as less successful counterparts: lessons learned?
- Note: lessons learned, not “history” *per se*

Some things that worked:

(Question discussed: characteristics of Internet protocols, economics, standards, deployment that were most problematical, might limit future success?)

- Incremental Permissionless Innovation
    - The End-to-End Arguments as interpreted by Larry Lessig[1] – the ability of any N people to create and deploy new applications and new uses without a) general agreement first or b) the permission of gatekeepers

- Architected Decoupling: Effectively positioned “fixed points” around which different parts of the system could evolve independently. (aka “the narrow waist”)
    - Extraordinary Lifetime: overall architecture survived and survives over many generations of technology and applications
    - Extraordinary Diversity: the basic design accommodated both technologies and uses never imagined of when initially created
- A corollary: Open Source and the Software Network Effect
    - Early recognition that giving away the key shared elements was more powerful than any proprietary approach

1: Lemley, Mark A. and Lessig, Lawrence, The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era (October 1, 2000). http://dx.doi.org/10.2139/ssrn.247737

Something that ~~didn't~~ doesn't, quite..

(Question discussed: characteristics of Internet protocols, economics, standards, deployment that were most problematical, might limit future success?)

Famous Economist Who Wishes To Remain Anonymous:

> *"The problem is that you people thought your job was to route packets. You didn't understand that your real challenge was to route money"*

An ongoing challenge to the Internet today:

> Today's news: *"Europe's biggest telecoms companies have called on the EU to compel Big Tech to pay a "fair" contribution for using their networks…"* [1]
>
> In the US: *Continuing stress around net neutrality, common carrier status, etc.*

Key Point: much more of an issue in an end-to-end IP-style architecture than in a more core-services-rich architecture, because critical "fixed points" in the system are definitionally reduced to commodities.

A fundamental example of tussle between economic and technical realities, and between the economics of the overall system vs the economics of (some) individual players.

1: https://telecoms.com/524024/european-telcos-once-again-push-for-fair-contribution-to-network-costs/

If you could change one thing about the Internet Protocol designs, what would that be?

I would have thought more about lifecycle.

- When a particular application or class of application is new, ease and flexibility of deployment is a dominant concern. Rapid evolution of the application itself is likely another.
- As an application or class matures, things change. Optimization, efficiency, security and stability become the dominant concerns.

Failure to have recognized and accommodated lifecycle leads us to things like “everything over HTTP” and “Sorry, new DNS record types are impossible now.”

An architecture that (somehow) explicitly accommodated lifecycle would do a better job for its users and face much less pressure for distortion over time.

Characteristics of Internet Protocols, Economics, Standards, and/or Deployment that best captures why they were so successful

- **Technical:** The minimal guarantees that allowed the interconnection of disparate (and at the time rapidly evolving) communication technologies vying for dominance: Ethernet, ATM, optical rings, cellular wireless.

- **Economic:** the “parent pays” model that funded both rapid development and free or nearly free connection to the net until commercially viable applications like eCommerce emerged.

- **Regulatory:** Avoidance of regulatory capture by the de jure standardization community in the CCITT and later ITU, at the same time as important telecom deregulation activities.

To ponder: Were these a combination unique in time and space and unlikely to recapitulate to accommodate coming disruptive changes?

Characteristics of Internet protocols, economics, standards, deployment that were most problematical, might limit future success?

- **Technical:** The ossification of the protocol suite along with the inability to bypass stateful intermediaries serious limits future design choices and locks in inefficiencies that may keep things from tracking communication advances

- **Economic:** the perceived need for massive scaling in order to be taken seriously, along with the rapid centralization of both Internet infrastructure and application services

- **Regulatory:** The "maturation" of the IETF which may overvalue "do no harm" over fast enough innovation

To ponder: The research community got funded to try "clean slate" designs which were super-cool but went nowhere. They didn't really influence the existing protocols in a meaningful way either. Is there an alternative approach?

If you could change one thing about the Internet Protocol designs, what would that be?

My personal favorite: Un-invent GetHostByName and other Berkeley sockets calls that exposed IP addresses to the application layer. This has had a number of very serious negative effects:

- A fundamental cause of why IPv6 took 10+ years to roll out
- Exposes multi-path and connection rebinding above the transport layer making these capabilities very hard to deploy (e.g. MP-TCP)
- Greatly expands the attack surface by making reflection attacks trivial

General principle to ponder: Be way more careful about layer violations - they can come back and bite you many technology cycles later

How did we get to various architecturally ugly places?

As John and Dave have suggested in their talks, much of the seminar looked at how we had ended up in various architecturally ugly places and could we, perhaps, have done better.

I'll briefly touch on two that interact to limit innovation

- **Security**
- **IP addresses**

Security

- **Data communications security was evolving concurrently with the Internet:** understood encapsulation and essentials of public-key when TCP/IP was developed, but security associations and per-packet encryption rules came later (and remember US munitions laws got in the way too!)
- **Firewalls:** a known problem (Dave Mills was coining terms for all the misbegotten packets he saw with his packet printer) that the community failed to focus on. Result was we got firewalls - boxes that sought to enforce correctness of packet formats and limit which packets got to cross boundaries. Layer 3 and 4 protocol innovation died with firewalls and the network effect (aka NxM rule) makes it unlikely it will come back

IP addresses too small

- **IP addresses were too small:** and there were others thinking bigger at the time (e.g. Ethernet addresses were 6 bytes long). So we were forced into Network Address Translation (NAT)
- **Alas, we also allowed IP addresses to become first class objects:** with the result that applications put addresses in their content. NAT and firewalls had to go scraping through content to fix those addresses as they transited middleboxes. Middleboxes that do scraping make it hard to add new protocols (which might have IP addresses embedded in new ways…)

The homework assignment test

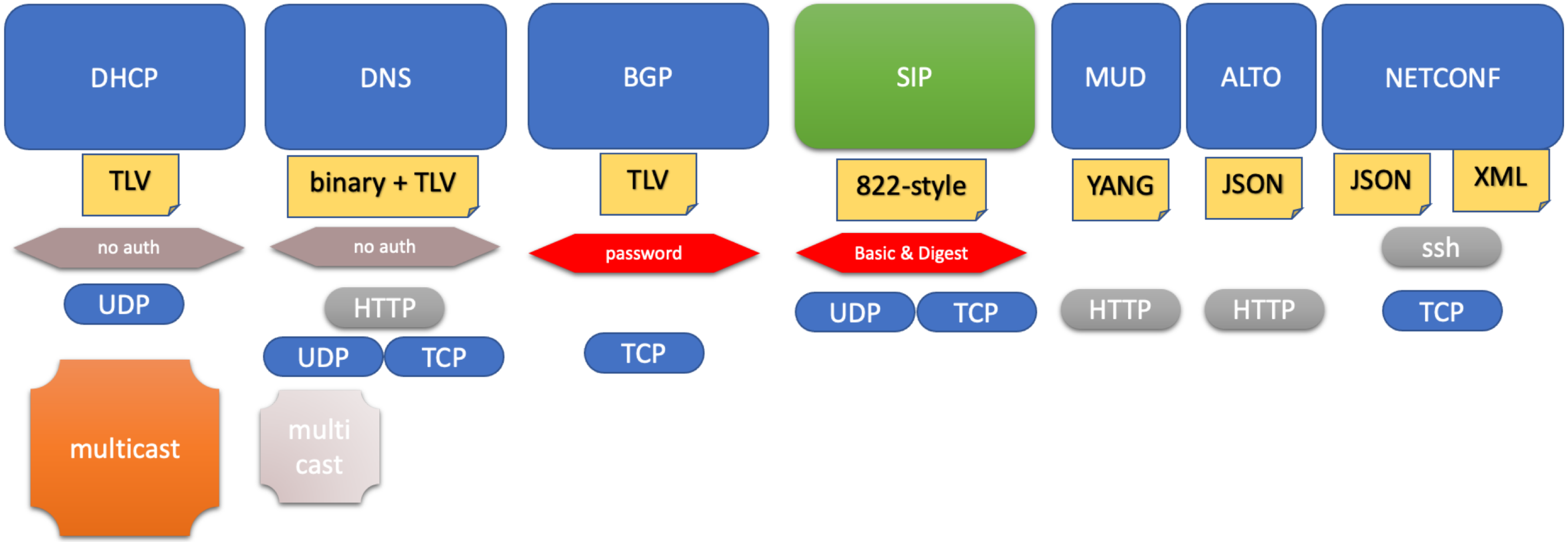
- Many internet protocols *in their initial specification* are simple enough for a senior-level CS homework assignment
- can implement interoperable subset (i.e., the introductory example)
- e.g., IPv6, RIP, UDP, TCP, DNS, HTTP/1.1 (subset), SMTP, SIP, …
- ~hundreds-1,000 LoC
- But many protocols are specified by a patch model (“updates RFC…”)
- But building a real “industrial-strength” protocol requires probably 100 times that much
- very small number of full implementations (2-3, maybe)
- The liberal part of “Postel’s Law” increases security-relevant surface and state space
    - Parsing is most likely CVE
    - Encourages retaining legacy code to reduce test effort

Software ate interoperability

- The only mostly compatible applications are email, web (rendering) and calendaring (as email attachment)
- WebRTC does not count – no semantic-level access
- SIP was the last user-facing application layer protocol standardized in the IETF
- Regression compared to prior generation of communication technology
  - high level of functional compatibility for basic phone (and fax) services from 1900's through 2010's (even on mobile)
- enabled by software implementation and auto-installation – one classical motivation for standards disappears
  - with the web browser as the ultimate auto-install tool
- fundamental shift in economics and "minority protection" (5% could force compatibility – now gets told to install software or use web client)
- identify clearly what we're losing – is it worth the effort to fight?
- may require regulatory intervention on anti-trust grounds

If I could change one thing: a capable control layer, with assured identity

- Our control protocol stacks are examples of non-reuse
  - DNS, DHCP, routing, network management, configuration, …
- discovery, reliability, confidentiality, integrity, authentication, error management, observability, …



Connectivity: Whom is the Internet for?

- We still assume IP nodes to always be connected (to infrastructure)
  - in spite of mobility, unstable Internet, hyperlocal networks, …
  - Extra(?) burden on L1/L2 (and L3) infrastructure
  - Some protocol support for changing IP addresses (QUIC) or independence of transport connections (HTTP, SIP)

- and to be connected at high performance (low RTT, high data rate)
  - kind of abandoning the lower end of the scale

- Many applications perform rather poorly outside their “favorable” conditions
  - Driven by content, appearance, service, … demands
  - Adaptation only carries so far

IP Evolution: What is / will be left of IP?

- Transparent end-to-end IP connectivity seems long gone (cf. middleboxes)
- Internet “backbone” vs. CDNs and hyperscalers w/ their own global networks
- User interactions happen mostly endpoint to cloud
  - Replicated cloud instances close to the respective user hosted by CDNs
- IP (and UDP/TCP) as just an API for the access link/path?
  - Instead of global reach
  - Everything else defined by / within the application
  - Opportunity to re-enable Internet innovation?*
- Implications for services and contents?
  - Fragmentation? Different local views? Censorship? …?
  - Transitivity: sending a link to somebody elsewhere to get the same result?

* E.g., Balakrishnan et al.: Revitalizing the Internet by Making it Extensible, ACM CCR, April 2021.

Beyond technology

- Internet technology doesn’t really define / limit its (future) usage
  - Still hard to do just “mechanisms and not policies”
  - (How) could one have better considered social and economic implications?

- Centralization
  - Internet protocols are (mostly) inherently distributed
  - Yet many facets of today’s deployment aren’t
- Content
  - Democratization vs. disinformation
- Governance
  - Hasn’t scaled up with Internet size and importance (and speed of development)

theNetworking Channel

Next event: Green Digit

Gergely Sipos – EGI

Yuri Demchenko – University of Amsterdam

Raffaele Bruno – CNR

Tian Guo – Worcester Polytechnic Institute

Organiser: Serge Fdida – Sorbonne Université