Architectural Principles of the Internet

IPAM Tutorial

March 12, 2002

Bob Braden

USC Information Sciences Institute

Marina del Rey, CA

Outline

- A brief historical overview of the Internet
- What is network architecture?
- The requirements for the Internet
- Architectural principles of the Internet
- Conclusion

Internet History: The Yellow Brick Road (1)

- 1969: Birth of the ARPAnet packet-switched network.
  - First IMP installed at UCLA
- 1974: Cerf/Kahn paper on internetworking
  - Many elements of the final Internet protocol design
- 1977: ARPA research program on internetworking
  - Important players included: BBN, DCEC, ISI, MIT, SRI, UCLA
  - 6 prototype implementations of TCP/IP

The Yellow Brick Road (2)

- Jan 1, 1983: Birth of the Internet
  - ARPAnet switched to TCP/IP protocols; Mil Std.
- ~ 1985: NSFnet
  - Major growth engine: general academic usage
- ~1989: Privatization of the Internet
  - People willing to pay to use it
  - People wanting to make money supplying services
- ~1991: World-Wide Web introduced
  - From academic tool to popular culture

The Good Witches

- DoD chose TCP/IP as Mil Std protocol ~1983
- CSNET chose TCP/IP ~1983
- ARPA directed Berkeley (BSD) Unix developers to implement TCP/IP.
- Courageous technical managers in DoD, NASA, DoE, and NSF supported TCP/IP *(FRICC)*
- NSF chose TCP/IP for building NSFnet

The Bad Witches (1)

- X.25
  - Common carrier packet switching service.
  - This Bad Witch captured the UK academic world ~1980.
  - Context: TCP/IP was tainted by the US DoD, and in 1982 there were 600,000 anti-nuclear demonstrators in London.
- ISO Open Systems Interconnect (OSI)
  - A developing international standard protocol suite.
  - A bureacratic dream and a technical nightmare.
  - US DoD tried to mandate OSI to replace TCP/IP at Mil Std.
  - As late as 1993, some believed OSI would win.

More Bad Witches (2)

- FAX
  - It seemed that FAX would kill email.
- PTTs (government monopoly telcos) in Europe and Asia
  - They didn’t get it, but they blocked Internet progress outside the US for a long time.

More Bad Witches (3)

- US telco never *got it*.
  - They had built the hugely successful telephone network and could not imagine any other reality.
    - Very sophisticated engineering to solve one very specific and well-characterized communication problem.
  - They thought they were the *Wizard of OZ*.
- ATM
  - The telcos [re-]invented packet switching in the form of ATM, and some thought it would rule the comm world.

Why did the Internet get to Oz?

- Some good luck and some clever moves.
  - Biggest factor: the Internet worked!
- ARPAnet research community mindset:
  - Driven by pragmatics (instead of dogmatics)
    "Rough consensus and running code"
  - Reductionist thinking

    Scientific viewpint, not engineering.

    Led to the: **Internet architecture *(IA)***

What is *Internet* **Architecture**…?

- A conceptual metaphor

Network Architecture

"A set of principles and basic mechanisms that guide network engineering."

- Boundaries fuzzy: bounded from "above" by requirements and from "below" by engineering.
- Even fuzzier: boundary between architecture and mechanism.

Historically, informal architectural ideas guided design of the Internet protocols, but the architecture was formalized later...

- *"The Design Philosophy of the DARPA Internet Protocols"*, David D. Clark, SIGCOMM '88, p.106.

Network Architecture

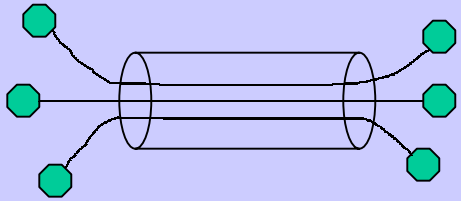
- What *entities* are named, and how?
- How do *naming*, *addressing*, and routing inter-relate?
- Where & how is *state* installed and removed?
- How are functions *modularized?*
- How are *resources* allocated?
- How are security *boundaries* drawn and enforced?

Primary (original) Requirements

- Multiplexing
- Survivability (robustness)
- Service generality
- Diverse network technologies

Multiplexing

- Basic issue: how to send multiple independent data streams across one physical channel? E.g.,
  - FDM Frequency-division multiplexing?
  - TDM Time-division multiplexing?
  - Packet switching?



Survivability (Robustness)

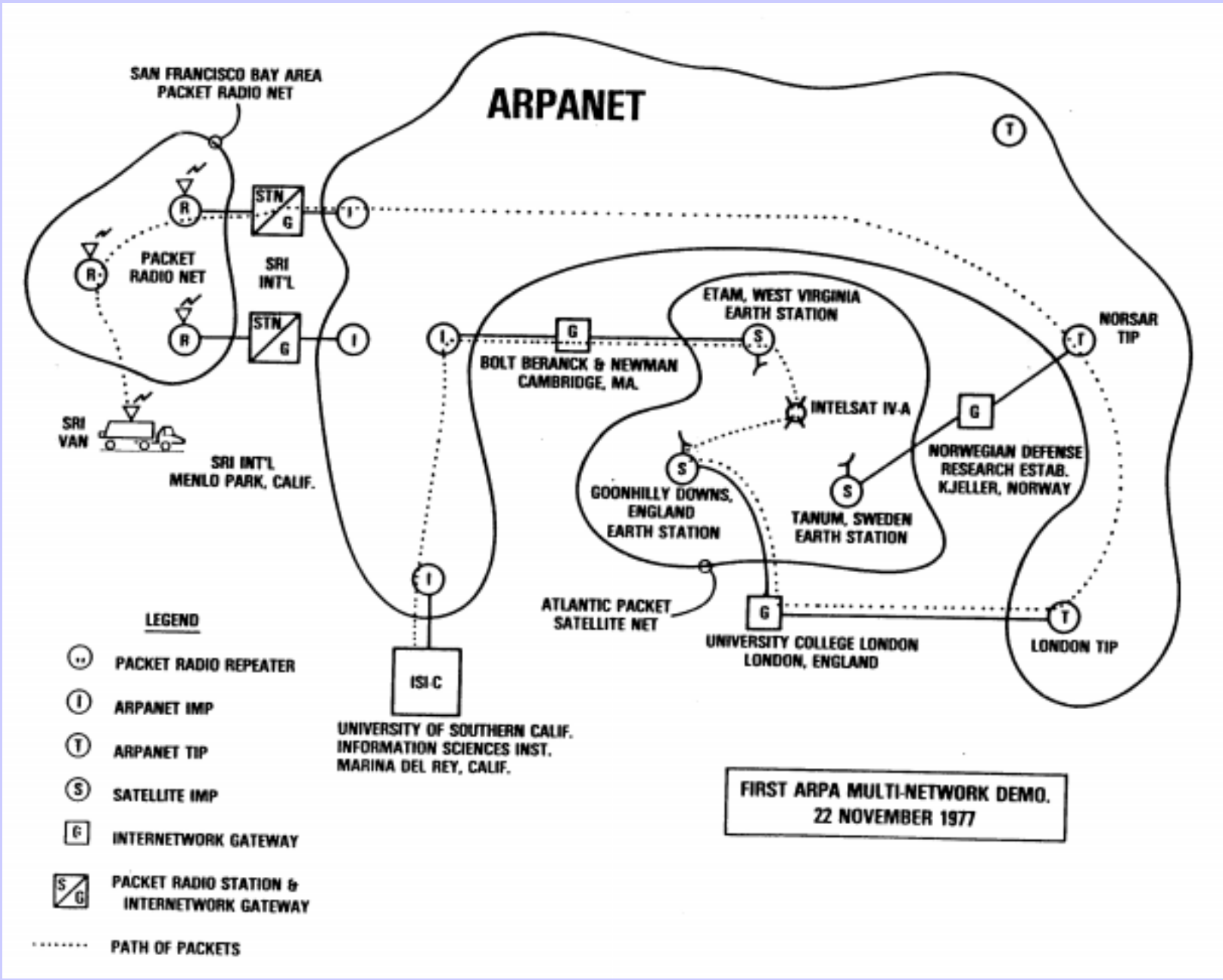
- This requirement was a Big Deal for a military-funded effort
  - Very high priority requirement: *messages get through no matter what, despite'very bad' things happening...*
  - Irony: survivable protocols are a boon in peace time -- we call it *robustness*.
- Implies dynamic adaptation to outages. May also imply protocols that are in some sense "self healing".

Service Generality

- Support widest possible set of applications.
- Support a range of communication service models
  - Virtual circuit service
    - Reliable, ordered, full-duplex data streams
  - Datagram service
    - Unreliable, unordered (“best effort”) service
  - (Isochronous service -- not a requirement)

Diverse Network Technologies

- Existing network (“subnet”) technologies
  - ARPAnet, Milnet (DDN)
  - Packet satellite networks (SATNET)
  - Packet radio networks (mobile/wireless)
  - LANs -- bus & token ring
  - Serial lines
  - X.25
  - Frame Relay
  - ATM
  - Sonet
  - WDM


ARPANET
SAN FRANCISCO BAY AREA
PACKET RADIO NET
PACKET
RADIO NET
STN
G
SRI
INT'L
SRI
VAN
SRI INT'L
MENLO PARK, CALIF.
BOLT BERANCK & NEWMAN
CAMBRIDGE, MA.
ETAM, WEST VIRGINIA
EARTH STATION
INTELSAT IV-A
GOONHILLY DOWNS,
ENGLAND
EARTH STATION
TANUM, SWEDEN
EARTH STATION
NORWEGIAN DEFENSE
RESEARCH ESTAB.
KJELLER, NORWAY
NORSAR
TIP
ATLANTIC PACKET
SATELLITE NET
UNIVERSITY COLLEGE LONDON
LONDON, ENGLAND
LONDON TIP
ISI C
UNIVERSITY OF SOUTHERN CALIF.
INFORMATION SCIENCES INST.
MARINA DEL REY, CALIF.
LEGEND
PACKET RADIO REPEATER
ARPANET IMP
ARPANET TIP
SATELLITE IMP
INTERNETWORK GATEWAY
PACKET RADIO STATION &
INTERNETWORK GATEWAY
PATH OF PACKETS
FIRST ARPA MULTI-NETWORK DEMO.
22 NOVEMBER 1977

Diverse Network Technologies

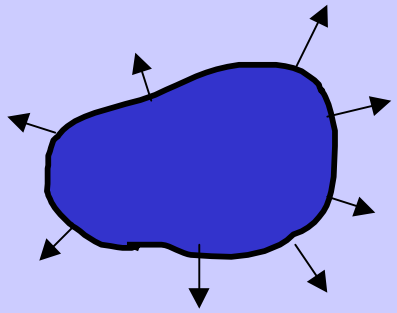
- Wide range of characteristics
  - Geographical extent, delay, bandwidth, errors, MTU (maximum transmission unit), broadcast? NBMA? etc.
- Implication 1: Need a network of networks.
- Implication 2: Heterogeneity is unavoidable!
  - *Despite industry hype for the latest "universal" technology ... X.25, ISDN, ATM, and now optical.*

Secondary/Later Requirements

- Scalability
- Distributed management
- Security
- Mobility
- Capacity allocation

Scalability (1)

- Growth dimensions:
  - N = number of end points (hosts)
  - Bmax = max bits per second
  - Dmax = max E2E delay in seconds



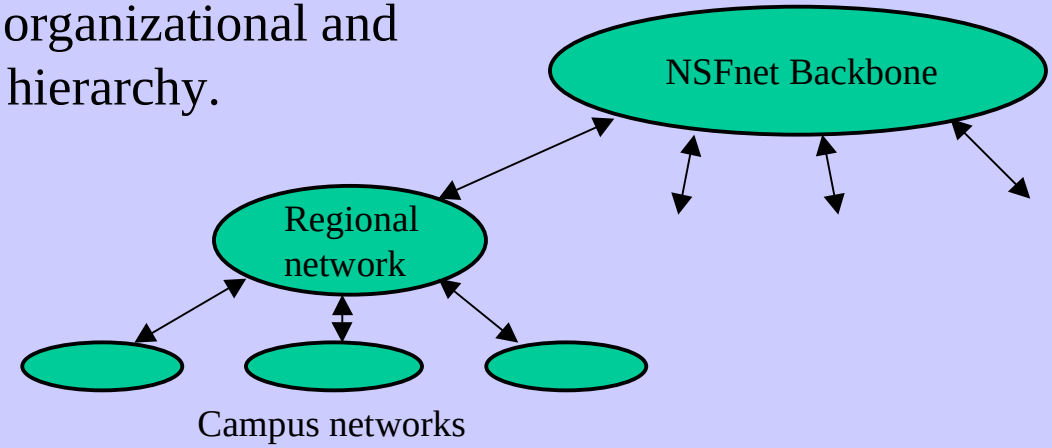
- Continuing exponential growth in N was unplanned! => Recurrent growth crises.
  - "Protocol X does not scale" usually means: total overhead ~ O(N), rather than O(log N).
  - In exponential world, log N => linear growth of cost.

Scalability (2)

- Bmax, Dmax:
  - Rate of growth of Bmax was also a surprise, although we have handled this better than N growth.
  - What matters is B*D product ~ bits in flight.
  - Must handle dynamic *ranges* 0:Bmax, 0:Dmax
  - Handling these is a mechanism problem, not an architectural problem.

Distributed Management

- Many administrative domains
  - After 1983, there was no time when entire Internet was under one management.
    - Early Internet: BBN did provide all routers (LSI/11s) and manage much of the Internet.
    - NSFnet => organizational and topological hierarchy.

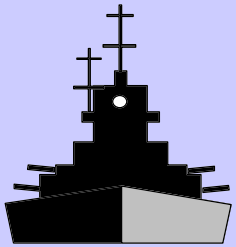


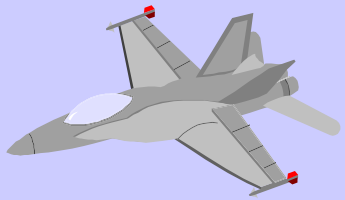
Security

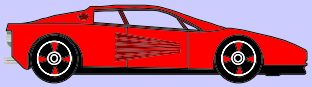
- All the usual stuff ...
  - privacy
  - integrity
  - authentication
- Strangely, security was a late addition to the architecture.
  - Military assumed link encryption (encrypting all traffic on each circuit.)

Mobility

- Mobility has been a secondary requirement
  - Localized solutions -- e.g., mobile IP -- grafted onto the side of the architecture.







Capacity Allocation

- Fairness
  - Weak requirement -- fairness was a social good, but it was at most a secondary requirement.
- Unfairness

  "Some pigs are more equal than others"
  - Early:DoD -> precedence hierarchy (IPv4 bits)
  - Today: ISPs want to sell different service qualtities for a price, and some users are willing to pay for better QoS (quality of service)

Non-Requirements

- Accountability
- Cost effectiveness
- Ease of host attachment
- Trust
- User empowerment

Accountability

- There has been no requirement to support usage accounting or $$ flows.
    - EG, a transit ISP cannot charge end user for better service.
- Whether or not this is a problem depends very much on your viewpoint.
  - Pro: Accounting -> service differentiation -> competition -> Good Things.
  - Con: Some (or all?) kinds of charging will distort usage and lead to moral decay.

Ease of Host Attachment

- It was tough in the early days, but really only a startup problem.
- Today every system comes with TCP/IP stacks, driver software, and NIC hardware.

Cost Effectiveness

- Engineers tended to hate the 20 bytes of overhead for IP or 40 bytes for TCP, but...
- The military did not care in the early days.
- The academic world did not care a lot during NSFnet days, since “daddy” NSF was paying.
- The miracles of silicon and glass have pretty much made this an non-issue, except in the “last mile”.

Trust

- Spammers, hackers, ... If you connect to the Internet, you are exposed.
  - Trust was not considered a goal -- until we lost it.
- Firewalls provide a (primitive) mechanism to recapture trust within ‘gated communities’, but they limit functionality.
- Maybe the ability to belong to multiple trusted sub-communities should be a future requirement.

User Empowerment

- Competition is a great driving force.
- If users could select ISP paths and if payment mechanisms exist (see *accountability*), then ISPs could be motivated to provide additional services
  - QoS
  - Multicasting
  - ...
- Maybe this kind of user empowerment should be a future requirement.

Requirements -- Summary

Today:

- Multiplexing
- Survivability/Robustness
- Service Generality
- Interconnect diverse networks
- Scalability
- Distributed management
- Security
- Mobility
- Capacity Allocation

Tomorrow?

- Trust
- User empowerment

But not necessarily in that order!

Internet Architectural Principles

I will divide these 15 principles into

- Fundamental principles
- Secondary principles

You may argue about this classification...

Fundamental Principles

P1. Multiplexing

P2. Transparency

P3. Universal connectivity

P4. End-to-End argument

P5. Subnet heterogeneity

P6. Common Bearer Service

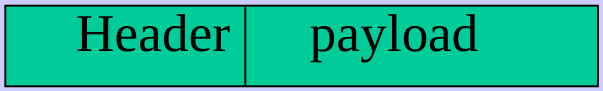
P7. Forwarding context

P8. Global addressing

P1. Multiplexing

P1.1. ***The Internet uses a single, global aproach to multiplexing: the variable-length*** **packet.**

- Self-contained --



- Header contains some *forwarding directive* (FD)
- Packet is universal unit for error detection & recovery.

P2. Transparency

P2.1. User data is delivered to the intended receiver without modification.

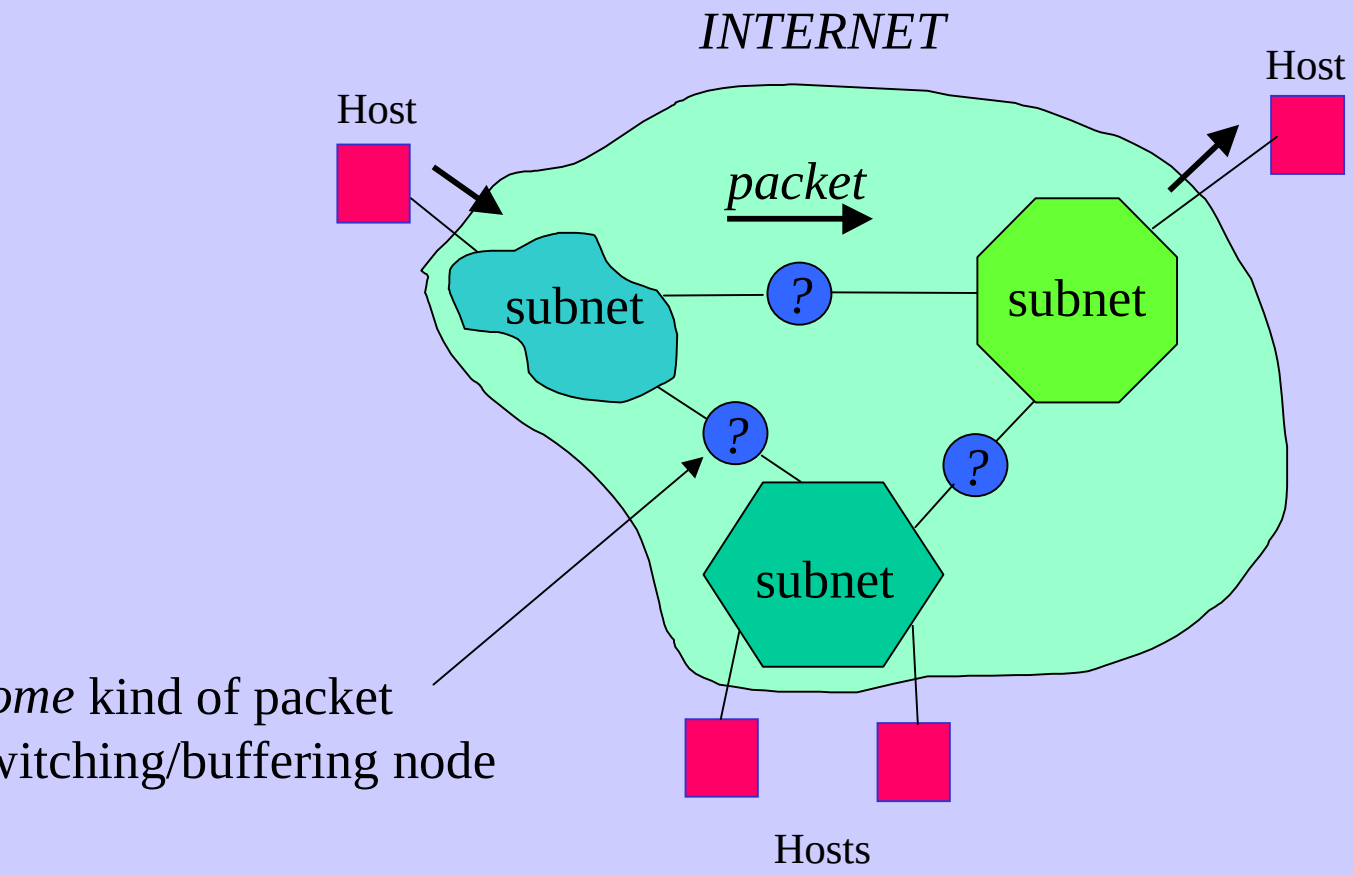
- The *"Don't mess with my data!!"* principle
- A controversial issue today, as ISPs start to mess with our data -- eg web caches that attach advertisements.

P3. Connectivity

P3.1. Any host can send packets directly to any other host (except when prohibited by policy).

- I.e., Internet communication is universal, direct and "real time", i.e., no indefinite delays.

P3.2. A host attached to any subnet of the Internet is "attached to the Internet."



Some kind of packet switching/buffering node

P4. “End-to-End Arguments” (1)

- The “end-to-end arguments” were presented in:
  - Saltzer, J., Reed, D., and D. Clark,
    “End-to-End Arguments in System Design”.
    2nd Int Conf on Dist Systems, Paris France, April 1981.
- Architectural reasoning about the appropriate location for communication functions -- network vs. end nodes.
- Wonderfully ambiguous, but often cited -- the closest thing to a sacred text for the Internet architecture.

P4. “End-to-End Arguments” (2)

P4.1. The network is built with no knowledge of, or support for, any specific application or class of applications.

P4.2. A function that can be entirely accomplished in an end node is left to that node, and the relevant communication state is kept only in that node.

Hence, a crash that loses communication state will be coincident with the loss of the communicating application -- *“fate-sharing”*.

P4. “End-to-End Arguments” (3)

- Example 1: Reliable delivery

  End-to-end reliability (timeout, retransmit) can be entirely accomplished by the end nodes. P4.2 => implement reliable delivery entirely in the end nodes (e.g., using TCP).

- Example 2: Network buffering

  Speed variations between end nodes can be entirely handled by end-node buffering and flow control. P4.2 => network has no buffering mechanism to deal with end-node speed variation.

P4. “End-to-End Arguments” (4)

- Example 3: Format conversion

  Any required format conversion can be entirely handled by end nodes. P4.2 => network has no mechanism for format conversion between end nodes (hosts).

P4. “End-to-End Arguments” (5)

- The first academic statement of the “dumb network, smart end system” idea for computer communication.
  - Contrary to the telephone network: smart networks, dumb terminals.
- Today the E2E principles P4.1, P4.2 are often broken
  - Firewalls, NAT boxes, web caches, web proxies, etc. do application-specific processing within the network.

  Hotly-debated issues within the Internet technical community.

Where we are so far

- We have:
  - P1. Multiplexing: Packets everywhere
  - P2. Transparency: ‘Don’t mess with my data’
  - P3. Connectivity: Universal, direct, realtime
  - P4. E2E argument -- prefer functions in end nodes
- We have not yet specified:
  - What state is needed for forwarding packets? Is connection setup required, or is it connectionless?
  - How are addressing and routing performed?

P5. Subnet Heterogenity (1)

P5.1. The Internet makes the minimum possible assumptions about subnet functionality, in order to operate over diverse subnet technologies.

P5.2. Internet protocol designers should be willing to forego some efficiency and even functionality in order to maintain this flexibility and universality.

P5. Subnet Heterogenity (2)

- Optimizations can be harmful if they reduce future adaptability.
- In an exponential world, optimization is often an exercise in futility.
- Note: “minimum” assumption does not mean “no assumption.”
  - TCP performance requires packet loss under a few percent.
  - QoS often needs support in subnets.

P6. Common Bearer Service

P6.1. A universal internetworking protocol IP forms a "common bearer service" end-to-end.

- IP packets are forwarded E2E through each subnet.
- Subnets are linked by IP packet switches called *routers*.
- The service model is loosely defined -- "best effort" -- to handle diverse subnet characteristics.
  - Packets may be dropped, duplicated, or reordered.

P7. Forwarding Context

The Internet is ***connectionless***, i.e.:

P7.1. No setup is required before sending a packet. Packets are self-contained within the context of a global routing computation.

P7.2. Routers contain no per-flow state.

P8. Global Addressing

P8.1. A single global address space identifies the network attachment points of nodes.

These *IP addresses* are carried in IP headers and used by routers for packet forwarding decisions (FDs).

P8.2. IP addresses are also used as node identifiers ("names").

If you know the IP "name" of a host, you also know its address.

Secondary Principles

P1. Multiplexing
P2. Transparency
P3. Universal connectivity
P4. End-to-End argument
P5. Subnet heterogeneity
P6. Common Bearer Service
P7. Forwarding context
P8. Global addressing

P9. Routing
P10. Regions
P11. Protocol Layering
P12. Minimal Dependency
P13. Security
P14. Congestion
P15. Resource Allocation
P16. Mobility

P9. Routing

P9.1. The Internet performs a globally-consistent routing computation to support loop-free, hop-by-hop forwarding of packets.

P9.2. This routing computation is distributed so there will be no single point of failure.

P9.3. Source routing is supported as an exception to allow delivery when the routing computation does not provide an effective route.

P10. Regions

P10.1. The Internet supports administrative regions (domains) for the purpose of routing.

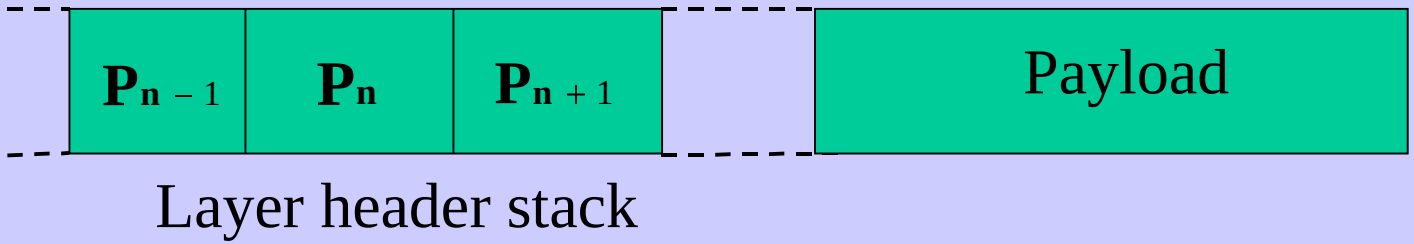
- A two-level hierarchical routing computation is performed, within and among regions.

P11. Layered Protocol Stack (1)

P11.1. Communication protocols are defined using layered abstractions.

- Layer N presents a service to layer N+1, and constructs this service using only layer N-1.

P11.2. Layering is realized using a last-on, first-off stack of layer headers on each packet.

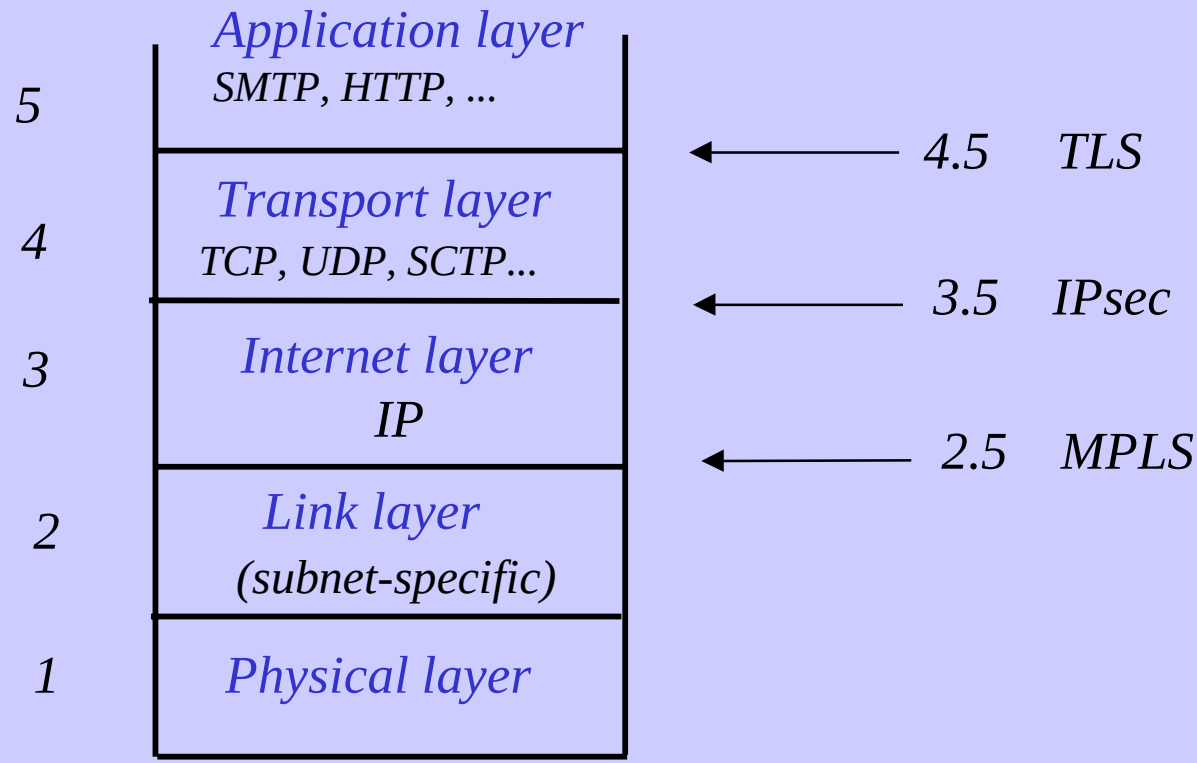


Layered Protocol Stack (2)

- Layering provides powerful model for building complex protocol interactions -- modularity, abstraction, information hiding.
- BUT the strict layering model is often violated.

  Thus, layer N's service may depend upon information at layer N+1.
- New functionality often requires new sub-layers

Internet Layer Cake



P12. Minimum Dependency

P12.1: A minimum of network services is required for end-to-end communication.

- ***Two Internet hosts that know each other's IP addresses can communicate without additional network support.***
  - I.e., the DNS is not required => robustness, boot-strapping.
- ***Two Internet hosts can communicate directly without an intervening router.***
  - Symmetrical interface -- no network access protocol.

P13. Security

P13.1. The Internet has an architected end-to-end security mechanism for integrity and privacy.

P13.2. Internet architecture has no mechanism to constrain hosts that offer excess traffic (DDos)

P13.3. Internet architecture has no specific mechanisms to defend its own elements from attack.

P14. Congestion (1)

P14.1. The Internet contains sufficient buffering to allow a host to run a congestion adaptation algorithm with a round-trip of control latency.

P14.2 The Internet provides an explicit congestion signal that can be used to drive adaptation algorithms.

P14. Congestion (2)

P14.3 Transport protocols should be no more aggressive than TCP. However, there is no enforcement mechanism.

P14.4. The network has no model of its own performance; a host must measure performance itself.

P15. Resource Allocation

P15.1. The Internet provides an architected mechanism for explicit network QoS for individual flows (int-serv)

P15.2. The Internet provides a mechanism to allocate network resources to aggregated flows (diff-serv).

P15.3 The Internet provides a "virtual path" mechanism for traffic engineering (MPLS)

P16. Mobility

- ***P16.1 The architecture optimizes for the stationary case; mobility support uses special case mechanisms (e.g., mobile IP, MANET) with extra cost.***

Internet Architectural Principles

P1. Multiplexing
P2. Transparency
P3. Universal connectivity
P4. End-to-End argument
P5. Subnet heterogeneity
P6. Common Bearer Service
P7. Forwarding context
P8. Global addressing
P9. Routing
P10. Regions
P11. Protocol Layering
P12. Minimal Dependency
P13. Security
P14. Congestion
P15. Resource Allocation
P16. Mobility

Conclusion

- We have certainly reached the emerald city, but we have not reached Kansas yet. (Remember what happen to Dorothy...)
- The Internet architecture is not “finished”.

Conclusion

Every one of the 16 architectural principle categories is problematic in some manner!

(a) Being broken for commercial reasons

(b) Being broken to obtain additional functionality

(c) Protected against unwise optimization only by constant struggle in the IETF.

(d) Represent real unmet requirements

(e) Mods suggested by researchers.

(f) Mods urged by mysterious government agencies

- Details? Need another 2 hours!

Acknowledgments

- Version -1 of this presentation began from a recent note by Dave Clark (MIT): “Principles”, Jan 22, 2002, as well has his 1988 paper. It also contains ideas from John Wroclawski and Karen Sollins of MIT. And of course it benefits from thousands of conversations with many, many Internet gurus over the years.
- This work was funded in part by DARPA under the NGI program.
  - See http://www.isi.edu/newarch