CS 678 Spring 2013
Network Architecture and Principles

“The Design Philosophy of the DARPA Internet Protocols”,
Dave Clarke, 1988

Ihsan Ayyub Qazi
Computer Science Department
LUMS SBASSE

Slides use info from Nick McKeown and Dave Andersen
Context: David D. Clark

- Senior Research Scientist at MIT
- Chief Protocol Architect for the Internet from 1981
  - Continues to be a network visionary today
  - NSF Future Internet Architecture Project

- At the time of writing (1987)...
  - (Almost) no commercial Internet
  - 1 year after Cisco’s 1st product, IETF started
  - Number of hosts reaches 10,000
Internet Architecture

• Fundamental goal:
  “Effective technique for multiplexed utilization of existing interconnected networks”

• Goals, in order of priority:
  1. Continue despite loss of networks or gateways
  2. Support multiple types of communication service
  3. Accommodate a variety of networks
  4. Permit distributed management of Internet resources
  5. Cost effective
  6. Host attachment should be easy
  7. Resource accountability
Fundamental Goal

• “Effective technique for multiplexed utilization of existing interconnected networks”
  – Specific reason: to give users on the ARPA radio packet network access to machines on the APRNET

• Multiplexing
  – Shared use of a single communication channel

• Interconnecting networks
  – Interconnect existing networks by defining a minimum set of requirements to support as many as possible
Techniques for Multiplexing

- Several techniques for multiplexing
  - TDMA, FDMA, etc
  - Statistical multiplexing (*Stat Mux*)

- *Stat Mux*: efficient sharing of resources
  - A link can *always* transmit when it has data!

- Packet Switching
  - Message broken down in ‘packets’
  - Packet forwarding info in dest. address of a packet
    - *No state* established ahead of time
  - Packet switched networks use *Stat Mux*
Discussion

• Q. Why did the Internet designers decide to interconnect existing networks rather than design a new unified network from scratch?
Discussion

• Q. Why did the Internet designers decide to interconnect existing networks rather than design a new unified network from scratch?

• Several Reasons:
  – High Cost
    • Different types of networks were already in place
  – Administrative Control Challenges
    • Networks represent administrative boundaries control e.g., LUMS, IBM, Facebook networks
  – Challenges in incorporating new kinds of networks
    • Need for updating end-hosts and network elements
Discussion

• Why was packet switching picked for multiplexing?

• Other choices?
Discussion

• Why was packet switching picked for multiplexing?
  – Nature of applications to be supported
    • e.g., remote login
    • traffic tends to be bursty, reservation is inefficient

• What were the other choices?
  – Circuit Switching: Signaling protocol sets up entire path (e.g., the phone network)
  – Virtual Circuits: Hybrid approach. Packets carry “tags” to indicate path, forwarding over IP
  – Source routing: Complete route is contained in each data packet
Goal-1: Survivability

• If network is disrupted and reconfigured
  – Communicating entities should be able to continue!
    • No higher-level state reconfiguration
  – Transport interface only knows “working” & “not working”
    • Network masks transient failures
    • Not working == complete partition

• To achieve this state information about the ongoing conversation must be protected
Discussion

• Where can communication state be stored?

• What is the fate sharing principle?
Discussion

• Where can communication state be stored?
  – Network routers
    • Need replication of state in several routers
    • Maintaining consistency incurs overhead and is challenging
  – End-hosts
    • No need for replication
    • Protects against all network failures
    • Easier to engineer

• What is the fate sharing principle?
Fate Sharing

- Lose state information for an entity if and only if the entity itself is lost
- Examples:
  - OK to lose TCP state if one endpoint crashes
    - NOT okay to lose if an intermediate router reboots
  - Is this still true in today’s network?
    - NATs and firewalls
Goal-2: Types of Service

- Applications have different requirements
  - Elastic apps that need reliability: remote login or email
  - Inelastic, loss-tolerant apps: real-time voice or video
  - Others in between, or with stronger requirements
  - Biggest cause of delay variation: reliable delivery!
    - Today’s net: ~100ms RTT
    - Reliable delivery can add seconds

- Original Internet model: “TCP/IP” one layer
  - First app was remote login...
  - But then came debugging, voice, etc.
  - These differences caused the layer split, added UDP

- No QoS support assumed from below
  - In fact, some underlying nets only supported reliable delivery
  - Hard to implement without network support
Goal-3: Varieties of Networks

• Interconnect the ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links...

• Minimum set of assumptions for underlying net
  – Minimum packet size
  – Reasonable delivery odds, but not 100%
  – Some form of addressing unless point to point

• Important non-assumptions:
  – Perfect reliability
  – Broadcast, multicast
  – Priority handling of traffic
  – Internal knowledge of delays, speeds, failures, etc.

• Much engineering then only has to be done once
So, how do you support them?

- Need to interconnect many existing networks
- Hide underlying technology from applications

Decisions:
  - Network provides **minimal** functionality
  - “Narrow waist”

**Tradeoff:** No assumptions, no guarantees
The “Curse of the Narrow Waist” 😊

• IP over anything, anything over IP
  – Has allowed for much innovation both above and below the IP layer of the stack
  – An IP stack gets a device on the Internet

• Drawbacks:
  – difficult to make changes to IP
  – But...people are trying (SDN, GENI,..)
  – Only a small amount of information available about lower levels. (e.g., wireless)
    • Significant optimizations possible → cross-layer
Goal #4: Distributed Management

• Independently managed as a set of independent “Autonomous Systems”
  – ISPs, LUMS, Etc.

• BGP (Border Gateway Protocol) connects ASes together
  – Acts as a glue
A problem: Management

“Some of the most significant problems with the Internet today relate to lack of sufficient tools for distributed management, especially in the area of routing.”

The Internet is now a hugely complex beast
- >18,000 constituent networks
- Routing tables with 1,000,000+ entries

Management and operational expenses becoming increasingly important
Local Actions, Global Consequences

• Youtube and Pakistan

Pakistan Telecom complied by trying to change the BGP entry for YouTube to direct its internet users to a page that said YouTube was blocked.

Unfortunately, the ISP announced the new route to upstream providers. The upstream providers didn’t verify the new route but accepted it and then passed it along, cascading the bad address around the net, until most everyone using the net on Sunday would have been directed to the Pakistani’s network block. The blunder not only took down YouTube, but also choked the Pakistani ISP

NEWS: Pakistan’s Accidental YouTube Re-Routing Exposes Trust Flaw in Net

NEWS: Insecure routing redirects YouTube to Pakistan
Goal #5: Cost Effectiveness

• Packet headers introduce high overhead
  – but so does circuit setup

• End-to-end retransmission of lost packets
  – Potentially wasteful of bandwidth by placing burden on the edges of the network
Goal #6: Ease of Attachment

- IP is “plug and play” Anything with a working IP stack can connect to the Internet (hourglass model)

- A huge success!
  - Lesson: Lower the barrier to innovation/entry and people will get creative (e.g., Cerf and Kahn probably did not think about IP stacks on phones, sensors, etc.)

- But....

Tradeoff: Burden on end systems/programmers
Goal #7: Accountability

• Huge problem
• Accountability and security
  – Huge problem
  – Worms, viruses, etc.
    • Partly a host problem
  – Authentication
    • Purely optional

• Accounting
  – Billing? (mostly flat-rate)
  – Inter-provider payments
Discussion

• Why resource accounting on the Internet is hard?
Discussion

• Why resource accounting on the Internet is hard?

• Resource accounting requires the ability to monitor user traffic characterized transport layer sessions/flows

• Internet routers are stateless (consequence of fate sharing)
  – No information about flows at the network layer
  – Forced to treat each datagram in isolation
Discussion

• If the primary goal of the Internet architecture was to design a 'secure' network. How would the Internet look like?
Discussion

• If the primary goal of the Internet architecture was to design a 'secure' network. How would the Internet look like?

• Provisions for resource monitoring
  – Identify and monitor flows
  – Ability to report any violations
Design wrapup

• **IP model:** Stat mux, datagrams, fate sharing, narrow waist
  – **Successes:** IP on everything!
  – **Drawbacks**...but perhaps they’re totally worth it in the context of the original Internet
  • Might not have worked without them!

“This set of goals might seem to be nothing more than a checklist of all the desirable network features. It is important to understand that these goals are in order of importance, and an entirely different network architecture would result if the order were changed.”
Next Time:
“Networking named content”

• Some questions to ponder:
  – What are the key design principles behind CCN?
  – How are they different from the Internet Architecture?
  – What problems does it address which the Internet Architecture could not? Does it address all issues?