Network Architecture

- General thoughts
- Ethernet architecture
- Internet architecture
What is Network Architecture?

- **Services and APIs**
  - Ethernet – datagram broadcast
  - IP – best-effort datagrams over internets
  - socket programming interface

- **Protocols**
  - IP, TCP, ARP, ICMP, DHCP, DNS, RIP, OSPF, BGP, . . .

- **Algorithms/Mechanisms**
  - OSPF, BGP, DNS name resolution
  - longest prefix matching, congestion control
  - packet classification, policy-based packet handling

- **Applications**
  - Telnet, FTP, WWW, overlay networks, . . .
  - not strictly part of architecture, but motivate and shape it

- **Usage patterns, operational procedures**
  - use of port numbers by applications
  - administration of addresses
Architectural Assumptions

- Driving assumptions often implicit
  - to understand architecture, need to make assumptions explicit
  - effectiveness can depend on validity of assumptions

- Technology assumptions
  - network bandwidth, processing capabilities of network elements
  - maintaining state in network elements is hard/expensive
  - wireless bandwidth is limited, wireless power is scarce

- Application assumptions
  - internet for accessing expensive computing resources
  - no one wants video conferencing

- Assumptions about user behavior
  - hosts locations don’t change
  - exponential packet length distributions adequately model reality
  - users will not abuse internet openness

- Poor assumptions can lead to poor design choices.
- Assumptions can become self-fulfilling prophecies.
- Validity of assumptions generally changes over time.
Expecting the Unexpected

- Successful networks grow and last a long time
  - if objective is success, plan for “unreasonable” growth
  - easy to under-estimate network usage and longevity

- Technology capabilities change
  - can only predict with confidence for short term (say 10 years)
  - best not to let near-term constraints limit future developments
  - at same time, must be feasible in short term to succeed

- Networks get used in unexpected ways
  - Murphy’s Law for Networks – if users can do it, they will
  - unexpected uses can be positive (web) or negative (DoS)
  - can constrain non-standard uses (e.g. telephone network) or encourage them (e.g. internet)
    - even constrained nets get stretched (modems, fax, blue box)

- Aggregate behavior can emerge in strange ways
  - flash crowds, fractal traffic characteristics
Understanding Motivations

- Evolution of public networks depends on many stakeholders with variety of motivations
  - Internet service providers
    » more customers and more "value-added" services
    » reduce costs (equipment, installation, support)
  - Equipment vendors (both systems and components)
    » promote interest in new features
    » use technology to drive down cost (production & development)
  - Application and higher level service providers
    » need network to reach users and deliver services
    » network services constrain applications and quality of delivery
  - Policy makers
    » respond to constituents, lobbyists, technocrats
  - Research community
    » fame and (occasionally) fortune
  - Users and consumer organizations
    » avoiding growth in costs, ensuring broad access
Design Principles

- Can offer useful framework to guide design decisions
  - help maintain consistency as network evolves
- Example: protocol layering
  - each protocol layer should provide service through well-defined interface, while concealing implementation details
  - to facilitate correct implementation and enable change
- Example: end-to-end argument
  - network should provide only those services that cannot be provided effectively by endpoints
  - to minimize network complexity, avoid limiting applications
- Design principles can become controversial
  - admit variety of interpretations (QoS and end-to-end principle)
  - purists and pragmatists often dispute their sanctity
- Changing conditions can challenge their validity
  - are firewalls a blatant violation of e2e-ism, or an inevitable response to deficiencies in the internet architecture? Or both?
Elements of Effective Architectures

- Utility of provided services and supported applications
  » must be useful and must be used
- Minimal barriers to usage
  » easy for application developers to understand and use
- Scalability
  » in number of endpoints – how big is big enough?
    - \( N \) per person? what about tiny smart devices (smart dust)?
  » in geographic scope – local, national, global, galactic
  » performance of network elements (links, routers, end systems)
- Adaptability
  » make effective use of new technology as it develops
  » don’t limit architecture to constraints of current technology
- External factors often determine success
  » IP succeeded in spite of design flaws
    - BSD Unix, NSF-net and web were key drivers in its success
  » FDDI had significant technical advantages, but not enough to overcome Ethernet market dominance
Role of Geographic Distribution

- Whole point of networks is to connect remote endpoints
- Fundamental impact of distance
  - speed-of-light delays and impact on interactive applications
    - for both data and control
  - collision detection in CSMA/CD
  - power consumption of wireless links as function of distance
- Widely distributed networks have distributed control
  - equipment owned by individuals and organizations
  - typically means local and variable control
  - effective operation of the whole requires cooperation
    - advisable to minimize aspects that require cooperation between organizations – especially if organizations are competitors
- Network connectivity
  - constrained by technology, geography, organizational boundaries
  - impact on how traffic flows, how failures affect communication and who makes money
Modularity in Network Architectures

- “architecture...defines how system is broken into parts & how those parts interact.” – from NewArch Final Report
- Layered models used to describe network protocols
  - useful for defining services offered by layers, and reasoning about correctness
  - but, layer boundaries often violated for performance reasons
  - some functions (e.g. net management) necessarily span layers
- Modules and interfaces define implementation units
  - enable different organizations to implement different parts
  - allow for multiple versions of given parts
- Interfaces create opportunities for new functions
  - NAT depends on IP packet format, use of port numbers in UDP and TCP and prevalence of client-server interaction
  - firewalls depend on application usage of port numbers
  - usage patterns can lead to implicit interfaces
Ethernet Architecture

- Designed to connect computers in building or campus
- Technology-driven architecture
  - passive coaxial cable
  - asynchronous access, synchronous transmission
  - broadcast medium
  - access using CSMA/CD
  - 10 Mb/s transmission rate with Manchester encoding
Technology and Ethernet

- Historical context in early 1970s
  - mainframe and minicomputer era
  - early personal workstations in research labs
- Objective to make interconnection simple
  - manufacturer-assigned addresses, broadcast-based delivery
  - no address administration, no routing
- Passive network cabling
  - requires minimal planning, allowing easy expansion
  - largely impervious to equipment failures
- Implications of technology choices
  - need distributed arbitration method – CSMA/CD
    - collision detection places limit on
      \[
      (\text{data rate})\left(\frac{\text{geographic reach}}{\text{minimum packet length}}\right)
      \]
  - asynchronous access, synchronous transmission
    - phase-locked loops in receiving circuits had to be “trained” to lock
      onto sender’s frequency
    - required 7 byte preamble before each frame for reliable operation
    - Manchester encoding used to help maintain frequency lock
Frame Format

- **Preamble** enables synchronization of receivers.
- **Start of Frame** marks end of preamble.
- **Address fields** identify source and destination.
  - globally unique addresses, assigned by manufacturer of interface cards in terminals
  - no location information provided by addresses
  - address field of all 1’s is defined as **broadcast address**
  - **multicast addresses** specified by 1 in first address bit
    - multicast packets distributed throughout spanning tree
    - host Ethernet interfaces can be programmed to receive packets with specific multicast addresses

- **Type field** identifies type of data carried in frame.
- **Padding field** guarantees minimum frame length required by CSMA/CD algorithm.
  - minimum of 72 bytes per frame of which 46 bytes can be data and 26 bytes are overhead.
  - minimum frame duration of 57.6 µs at 10 Mb/s

- **Cyclic Redundancy Check field** (CRC) provides error detection.
Technology Evolution

- Twisted pair and passive hubs
  - in 1980s, technology allowed Ethernet over twisted pair
  - offices already wired in hub-spoke fashion for telephones
  - Ethernet could use same or similar wiring
  - for large installations, easier to manage than coax

- Bridges and switched Ethernet
  - large Ethernets became congested
  - first bridges were two port devices that localized traffic on different segments
    - learned locations of hosts by observing traffic
    - time out routing table entries to enable movement
  - correct operation depends on absence of cycles
    - spanning tree algorithm developed to break cycles in wiring
  - switches evolved as multi-port generalization of bridges
  - no change to basic protocols or packet formats

- Higher speeds (100 Mb/s, 1 Gb/s, 10 Gb/s, 100 G?)
  - retain classical packet format
  - more efficient transmission – 8B/6T, 4B/5B, 8B/10B, ...
Protocol Extensions

- IEEE 802 standards developed to harmonize different LAN technologies to facilitate interoperability
  - for Ethernet, type field replaced with length field and LLC-SNAP header containing type inserted after length

- Virtual LAN extension
  - uses four byte VLAN header inserted after source address
    - first two bytes specify Ethertype for recognition of VLAN tag
    - second two bytes include 12 bit VLAN tag and 3 bit priority
  - switches can be configured to constrain routing according to VLAN header and/or add/remove header
  - VLAN tags define broadcast domains
  - overlay different logical spanning trees on physical network
    - enables expansion in overall network capacity

- Stacked VLANs and Carrier Ethernet
  - to enable customer VLANs to be carried across carrier networks, additional VLAN header can be inserted
  - part of industry push to extend Ethernet to WAN applications
Reflections on Ethernet

- Technical simplicity enabled inexpensive deployment while not inhibiting future extensions
  - no real provision for extensions, but no serious obstacles
  - evolution depended on subsequent innovations, not planning
- Open standard that displaced proprietary technologies.
  - enabled innovation by many companies
- Dramatic improvements in network bandwidth
  - original Ethernet took local nets from <50 Kb/s to 10 Mb/s
  - subsequent upgrades now deliver 10 Gb/s at <$1K/switch port
- Innovation in addressing
  - unprecedented to devote 12 bytes to addressing
  - eliminated most system administration
- Use of broadcast foundation a mixed blessing
  - simplicity key to initial success
  - scaling up requires VLANs and remains awkward
  - in switched environment multicast usage limited, since broadcast required
An *internet* is a “network of networks” in which *routers* move data among a multiplicity of networks.

- heterogenous network types, multiple admin. domains

The Internet use the *Internet Protocol* (IP).

- *datagram* protocol with variable length packets and structured addressing
The Internet Hourglass

- IP as the narrow waist.
- Diversity above IP
  - transport protocols
  - applications
- Diversity below IP
  - LANs – Ethernet, FDDI, ATM, 802.11,
  - physical medium – 10/100, OC-48, wireless
Design Principles (from NewArch report)

- Packets are fundamental unit of multiplexing
  - not circuits, not virtual circuits, not cells
- Transparency – what goes in, comes out
  - no format conversions or other processing by network
- Universal connectivity as default state
- Immediate delivery
  - continuous connectivity, no long-term storage
- End-to-end principles
  - generality – network knows nothing about applications
  - robustness – if end nodes can do something, it’s left to them
  - fate-sharing – loss of state information for specific flow should coincide with loss of application
- Loose semantics
  - best-effort delivery only – no performance guarantees
- Subnet heterogeneity
  - assume little, so can use any subnet technology (almost)
More Design Principles

- **Common bearer service**
  - best-effort, connectionless datagram service
  - exceptions: source routing, multicast, IntServ
  - no separate access protocol – e.g. no user-network interface

- **Connectionless network mechanism**
  - no per-flow state in routers
  - exceptions: multicast, IntServ

- **Global addressing**
  - globally unique addresses, hierarchically organized for routing
  - exception: NAT

- **Protocol layering**
  - provide modularity of functions – use “header stacking”
  - frequently violated in practice

- **Distributed control**
  - no single point of failure
Still More Design Principles

- Global routing computation
  - to support consistent, loop-free packet delivery using destination addresses

- Multiple administrative regions (domains)
  - two level routing computation – inter-domain & intra-domain
  - may add additional levels within domain using OSPF

- Mobility
  - optimized for fixed host locations
  - mobile hosts accommodated using compatible extensions

- Network security
  - protection from eavesdropping is left to end-systems
  - no protection from denial-of-service

- Resource allocation
  - end-systems should back-off in presence of congestion
  - network provides enough buffering for e2e congestion control
  - Internet provides for QoS through IntServ and DiffServ
  - no architected support for payment-for-service
Last and Perhaps Least

- Minimal dependency
  - a minimal set of features sufficient for end-to-end packet delivery
  - endpoints can communicate directly without intervening router
IP Packet Format (v4)

- **Version number** specifies the version of the IP protocol and determines packet format.
  - version 6 is similar to v4 but uses longer addresses
- **Header Length (HLen)** gives number of 32 bit words in header.
- **Type of Service (TOS)** field can be used to allow application-specific treatment of packets.
- **Fragmentation Identifier, flags and Offset** used for fragmentation and reassembly of IP packets.
- **Time-to-live (TTL)** specifies the remaining number of hops before packet should be discarded.
  - prevents infinite looping of packets
- **Protocol** used for demultiplexing at destination.
- **Checksum** for end-to-end error detection.
- **Address** fields specify source and destination.
  - hierarchical address structure, CIDR
- **Options** are rarely used but must be supported in complete IP protocol implementations.
- **TCP** adds 40 bytes more, including port numbers.

<table>
<thead>
<tr>
<th>Ver</th>
<th>HLen</th>
<th>TOS</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frag. ID</th>
<th>HLen</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data (variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
# Internet Architecture Review

- **Best-effort, unicast datagram delivery service**
  - least-common denominator
    - only service one can really count on
  - usage of some elements discouraged (fragments, options)
  - effective sender anonymity raises fundamental security issues

- **End-to-end transport services**
  - defined by TCP, UDP and socket interface
  - most applications use TCP's *connection-oriented* service
  - TCP congestion control has led to unintended dependencies
    - router buffers sized to accommodate TCP behavior
    - de facto requirement for in-order delivery, stable routes and links with low packet loss
    - routers preemptively signal congestion by discarding packets
    - expectation that non-TCP protocols be TCP-friendly
  - TCP’s use of port numbers
    - forces specific application style (servers listening on ports)
    - requires global administration of port numbers
    - allows network to identify application
Internet Architecture Review

- **Applications**
  - email – POP, IMAP, SMTP, web-mail
    - server-based architecture needed for storage since network does not support deferred delivery
  - WWW – http, html, xml, CGI, javascript, ...
    - html made publication easy, browsers made it attractive to users
    - combination triggered explosion in content creation
  - emerging (?) applications
    - voice – slow to develop due to poor QoS support, competing options
    - real-time video broadcast – multicast deployment obstacles
    - interactive video – poor QoS and limited market interest
    - is cheap bandwidth & better queueing enough to overcome hurdles?

- **Control and management – ICMP**
  - error reporting (destination unreachable, TTL expiration, ...)
    - includes original packet’s IP header + 8 bytes of payload
    - unplanned uses – traceroute, distributed measurement efforts
  - miscellaneous other uses
    - echo (ping), redirect, source quench, timestamp
Naming and addressing
- originally, IP addresses used to identify hosts – DNS added later
- DNS maintains (name, value) pairs of various types
- distributed management with extensive use of caching to speed up responses
- an organization’s name server can use DNS to enables dynamic load balancing across servers
- availability of root DNS servers and correctness of DNS records is critical to Internet operation
  - original approach statically configured records – correctness depends on trusted communication among system administrators
  - dynamic update mechanism raises security issues – who to trust?

Connection and address assignment – DHCP
- hosts discover location of DHCP server using broadcast
  - large installations require a relay agent in each network
- automates address assignment in local networks
- enables convenient handling of mobile computers
- enabled by separation of names from addresses
Internet Architecture Review

- Address space conservation – NAT
  » observation of TCP setup process, port number translation

- IP and Ethernet – ARP
  » automates location of host with given IP address
  » leverages broadcast feature of Ethernet

- Routing – RIP, OSPF, BGP
  » fully distributed route computation for robustness
  » less distributed computation may work better
  » shortest path routing makes it difficult to distribute traffic
  » BGP’s policy-based routing leads to suboptimal decisions and is difficult to stabilize

- Management – SNMP
  » defines management information for individual components and mechanisms to retrieve it
  » no consistent framework for managing network as a whole
Internet Architecture Review

- Mobility – mobile IP (v4 and v6)
  » compatibility requirement makes it inefficient, awkward to use
  » DHCP suffices for most common cases
    - big exception: wireless IP phones

- Multicast – IGMP, DVMRP, MOSPF, CBT, PIM, BGMP
  » plethora of approaches, limited deployment
  » scalability concerns for reverse path forwarding
  » little economic motivation for ISPs

- Reservation & QoS – IntServ, RSVP, DiffServ, SIP, RTP
  » remains incomplete, blocked by scalability concerns,
    association with multicast, inadequate handling of inter-domain issues,...
  » reservation may require pay-for-service model

- Traffic Engineering – MPLS/GMPLS, RSVP-TE
  » enable better management of carrier networks
  » RSVP used to signal label-switched paths
  » no provision for end-to-end signaling
Reflections on IP Architecture

- Secrets of its success
  - the Internet idea – diversity above, diversity below
  - Ethernet, Berkeley Unix, NSF-net, email, WWW
  - ferocious advocacy – IP vs. the world (OSI, ATM)

- What started out simple has become complex
  - many moving parts (some fairly complex), subtle interactions
  - routers need large set of mechanisms to implement full IP protocol suite – many are never used (probably a good thing)
  - open trust model problematical in large public internet

- Technical deficiencies yet to be adequately addressed
  - address space limitations
  - users’ inability to control traffic they receive
  - support for mobile devices
  - quality of service, multicast

- In many cases, real issue is deployment obstacles
  - cost to upgrade equipment
  - need for universal agreement
General Lessons

- Beware assumptions
- Successful networks become complex
  » diverse stakeholders, new requirements, scale, security
- Building a truly general-purpose network is hard
  » least-common denominator approach hard to sustain
  » unconstrained featurism leads to complex interactions, subtle dependencies and ossification
- The Internet idea is powerful, compelling, essential
  » key question is what lies at the narrow waist
- Impact of technology profound, but uneven
  » bandwidth becoming more plentiful
  » extensive processing possible, even at gigabit rates
- Importance of wireless and/or mobile endpoints
  » fixed nodes becoming a small minority of total
  » need addressing mechanisms better suited to wireless devices
- Security a key concern for public networks
  » should insecure nets remain an option?