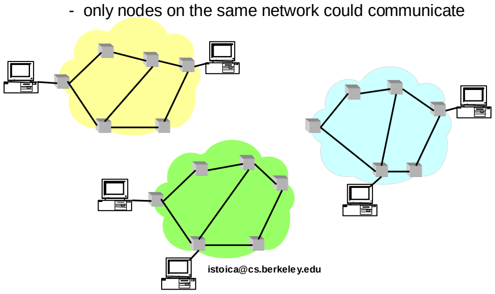
CS 268: Lecture 3 (TCP/IP Architecture)

Ion Stoica
January 28, 2003

The Problem

- Before Internet: different packet-switching networks (e.g., ARPANET, ARPA packet radio)
  - only nodes on the same network could communicate



Declared Goal

- "…both economic and technical considerations lead us to prefer that the interface be as simple and reliable as possible and deal primarily with passing data between networks using different packet switching strategies"

V. G. Cerf and R. E. Kahn, 1974

The Challenge

- Share resources of different packet switching networks → interconnect existing networks
- … but, packet switching networks differ widely
  - different services
    - e.g., degree of reliability
  - different interfaces
    - e.g., length of the packet that can be transmitted, address format
  - different protocols
    - e.g., routing protocols

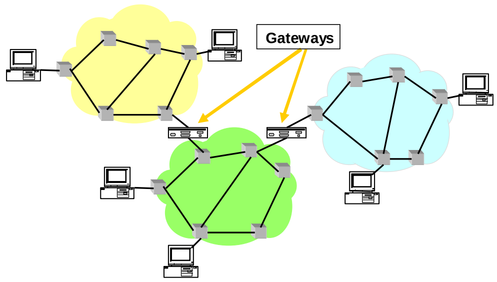
Possible solutions

- Reengineer and develop one global packet switching network standard
  - Not economically feasible
- Have every host implement the protocols of any network it wants to communicate with
  - Too complex, very high engineering cost

Solution

- Add an extra layer: internetworking layer
  - Hosts implement one higher-level protocol
  - Networks interconnected by nodes that run the same protocol
  - Provide the interface between the new protocol and every network

Solution



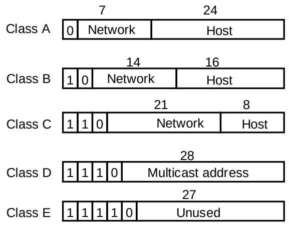
Challenge 1: Different Address Formats

- Map one address format to another. Why not?
- Provide one common format
  - map lower level addresses to common format
- Initially:
  - length: 24 bit
  - hierarchical
  - why hierarchical?

8	16
Network	TCP Identifier

Today Address Format (IPv4)

- Length: 32 bits
- Organization: hierarchical



Class A: 0 | Network (7) | Host (24)

Class B: 1 0 | Network (14) | Host (16)

Class C: 1 1 0 | Network (21) | Host (8)

Class D: 1 1 1 0 | Multicast address (28)

Class E: 1 1 1 1 0 | Unused (27)

What About the Future ?

- Internet is running out of addresses
- Solutions
  - Classless Inter Domain Routing (CIDR)
  - Network Address Translator (NATs)
  - Dynamic Address Assignments
  - …
  - IPv6
- Why not variable-sized addresses?

Challenge 2: Different Packet Sizes

- Define a maximum packet size over all networks. Why not?
- Implement fragmentation/re-assembly
  - Who is doing fragmentation?
  - Who is doing re-assembly?

Other Challenges

- Delivery time (propagation time + queueing delay + link layer retransmissions?)
- Errors → require end-to-end reliability
- Different (routing) protocols → coordinate these protocols

Service

- Unbounded but finite length messages
  - Byte streaming (what are the advantages?)
- Reliable and in-sequence delivery
- Full duplex

- Solution: Transmission Control Protocol (TCP)

Original TCP/IP (Cerf & Kahn)

- No separation between transport (TCP) and network (IP) layers
- One common header
  - Use ports to multiplex multiple TCP connections on the same host

32	32	16	16	8n
Source/Port	Source/Port	Window	ACK	Text

- Byte-based sequence number (Why?)
- Flow control, but not congestion control

Today's TCP/IP

- Separate transport (TCP) and network (IP) layer (why?)
  - Split the common header in: TCP and UDP headers
  - Fragmentation reassembly done by IP
- Congestion control (see next lecture)

IP Header

0	4	8	16	19 31
Version	HLen	TOS	Length	
Identification			Flags	Fragment offset
TTL		Protocol	Header checksum	
Source address				
Destination address				
Options (variable)				

20 bytes

- Comments
  - HLen – header length only in 32-bit words (5 <= HLen <= 15)
  - TOS (Type of Service): now split in
    - Differentiated Service Field (6 bits)
    - remaining two bits used by ECN (Early Congestion Notification)
  - Length – the length of the entire datagram/segment; header + data
  - Flags: Don't Fragment (DF) and More Fragments (MF)
  - Fragment offset – all fragments excepting last one contain multiples of 8 bytes
  - Header checksum - uses 1's complement

TCP Header

0	4	10	16	31
Source port			Destination port	
Sequence number				
Acknowledgement				
HdrLen		Flags	Advertised window	
Checksum			Urgent pointer	
Options (variable)				

- Sequence number, acknowledgement, and advertised window – used by sliding-window based flow control
- Flags:
  - SYN, FIN – establishing/terminating a TCP connection
  - ACK – set when Acknowledgement field is valid
  - URG – urgent data; Urgent Pointer says where non-urgent data starts
  - PUSH – don't wait to fill segment
  - RESET – abort connection

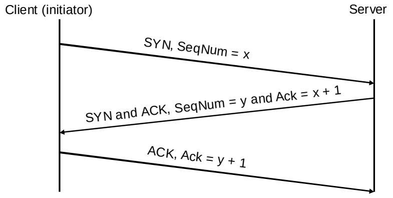
TCP Header (Cont)

- Checksum – 1's complement and is computed over
  - TCP header
  - TCP data
  - Pseudo-header (from IP header)
    - Note: breaks the layering!

Source address		
Destination address		
0	Protocol (TCP)	TCP Segment length

TCP Connection Establishment

- Three-way handshake
  - Goal: agree on a set of parameters: the start sequence number for each side



Back to the big picture

Goals (Clark'88)

0 **Connect existing networks**
  - initially ARPANET and ARPA packet radio network

1. Survivability
   - ensure communication service even in the presence of network and router failures
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

1. Survivability

- Continue to operate even in the presence of network failures (e.g., link and router failures)
  - As long as the network is not partitioned, two endpoint should be able to communicate…moreover, any other failure (excepting network partition) should be transparent to endpoints
- Decision: maintain state only at end-points (fate-sharing)
  - Eliminate the problem of handling state inconsistency and performing state restoration when router fails
- Internet: stateless network architecture

2. Types of Services

- Add UDP to TCP to better support other types of applications
  - e.g., "real-time" applications
- This was arguably the main reasons for separating TCP and IP
- Provide datagram abstraction: lower common denominator on which other services can be built
  - service differentiation was considered (remember ToS?), but this has never happened on the large scale (Why?)

3. Variety of Networks

- Very successful (why?)
  - Because the minimalist service; it requires from underlying network only to deliver a packet with a "reasonable" probability of success
- …does not require:
  - Reliability
  - In-order delivery
- The mantra: IP over everything
  - Then: ARPANET, X.25, DARPA satellite network..
  - Now: ATM, SONET, WDM…

Other Goals

- Allow distributed management
  - Remember that IP interconnects networks
    - Each network can be managed by a different organization
    - Different organizations need to interact only at the boundaries
    - … but this model doesn't work well for routing
- Cost effective
  - Sources of inefficiency
    - Header overhead
    - Retransmissions
    - Routing
  - …but routers relatively simply to implement (especially software side)

Other Goals (Cont)

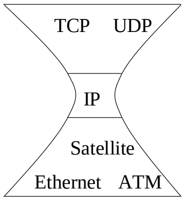
- Low cost of attaching a new host
  - Not a strong point → higher than other architecture because the intelligence is in hosts (e.g., telephone vs. computer)
  - Bad implementations or malicious users can produce considerably harm (remember fate-sharing?)
- Accountability
  - Very little so far

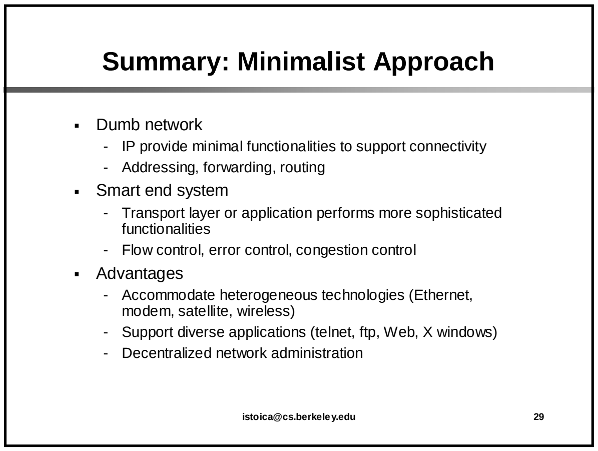
What About the Future?

- Datagram not the best abstraction for:
  - resource management,accountability, QoS
- A new abstraction: flow?
- Routers require to maintain per-flow state (what is the main problem with this raised by Clark?)
  - State management
- Solution
  - Soft-state: end-hosts responsible to maintain the state

Summary: Internet Architecture

- Packet-switched datagram network
- IP is the glue
- Hourglass architecture
  - All hosts and routers run IP
- Stateless architecture
  - No per flow state inside network





Summary: Minimalist Approach

- Dumb network
  - IP provide minimal functionalities to support connectivity
  - Addressing, forwarding, routing
- Smart end system
  - Transport layer or application performs more sophisticated functionalities
  - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration