CS 268: Lecture 3 (TCP/IP Architecture)

Kevin Lai and Ion Stoica

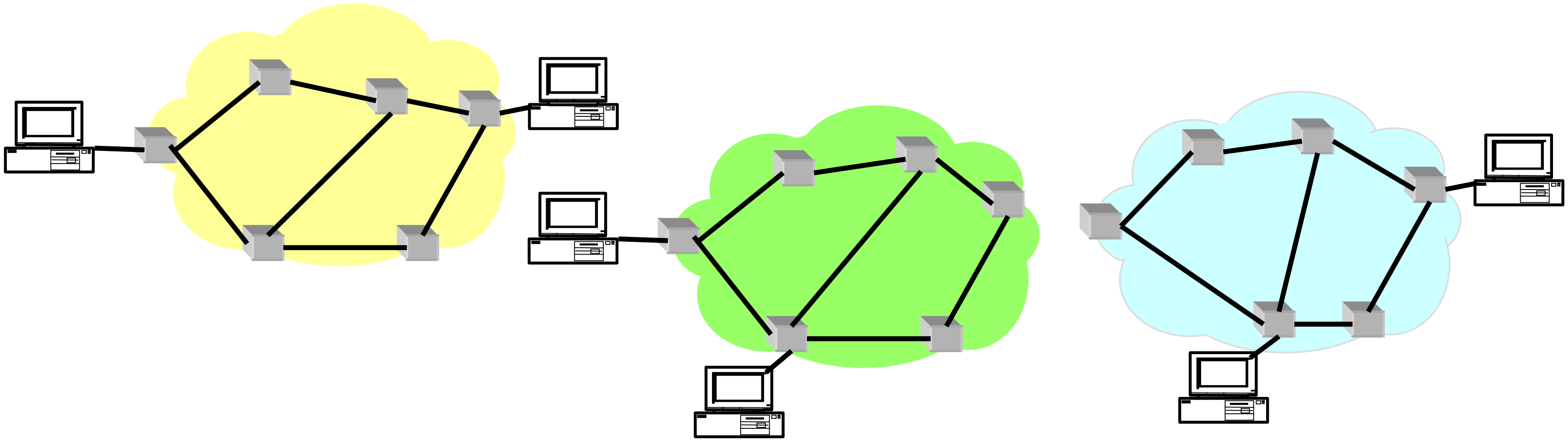
January 30, 2002

Paper Reviews

- state
  - goal of the paper
  - approach the paper takes to accomplish that goal.
- critique the paper by stating and justifying your opinion of the paper's
  - motivation
  - relevance
  - analyses
  - experiments

The Problem

- Before Internet
  - different packet-switching networks (e.g., ARPANET, ARPA packet radio)
  - only nodes on the same physical/link layer network could communicate
  - want to share room-size computers, storage to reduce expense



The Challenge

- 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

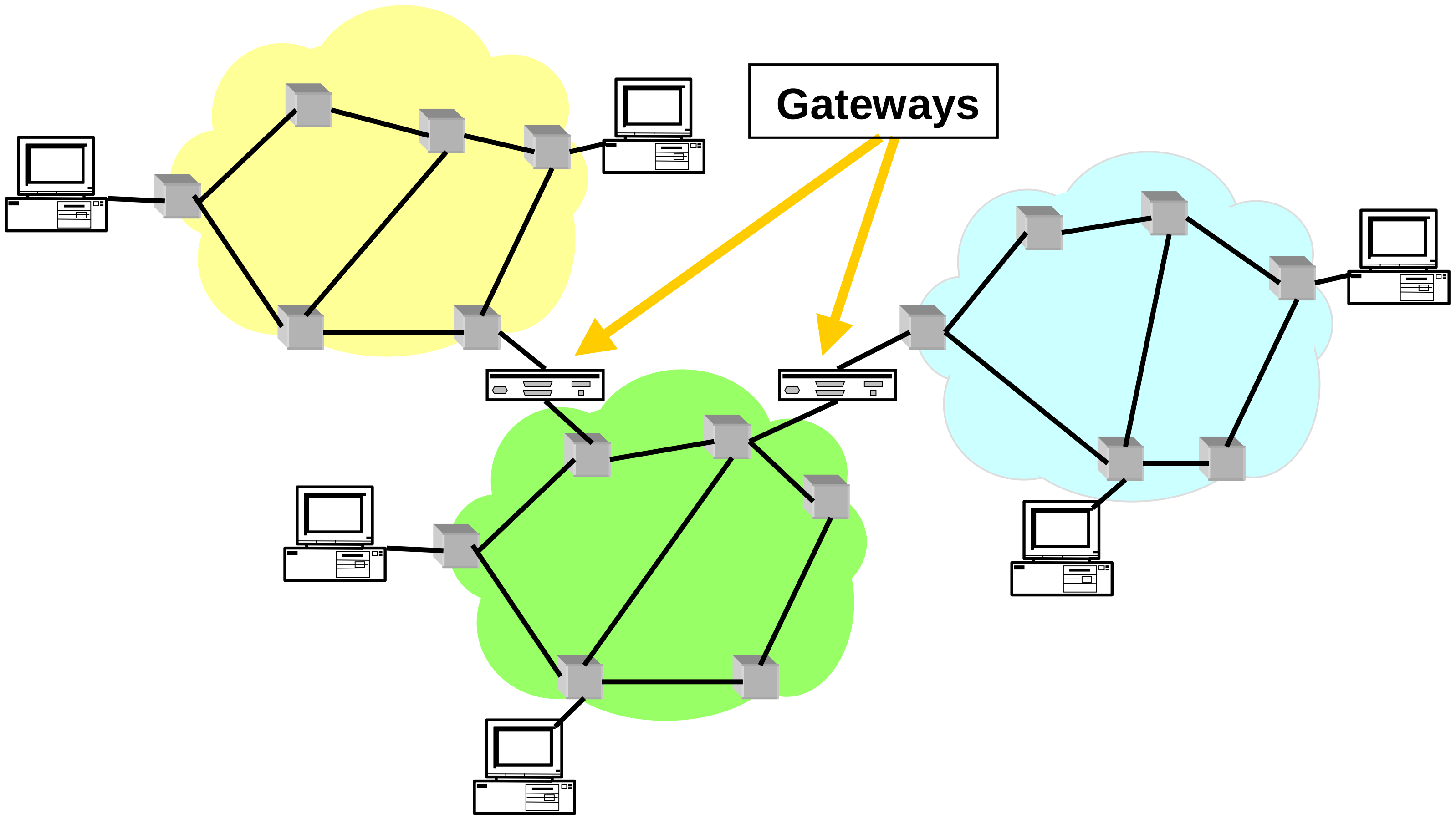
Possible solutions

- Reengineer and develop one global packet switching network standard
  - not economically feasible
  - not deployable
- Have every host implement the protocols of any network it wants to communicate with
  - Complexity/node = O(n)
  - $O(n^2)$ global complexity

Solution

- Add an extra layer: internetworking layer
  - hosts:
    - understand one network protocol
    - understand one physical/link protocol
  - gateways:
    - understand one network protocol
    - understand the physical/link protocols of the networks they gateway
  - Complexity to add a node/network: O(1) with respect to number of existing nodes

Solution



Common Intermediate Representation

- Examples:
  - telnet, IP, strict HTML, I-mode cHTML
- Who ignored this:
  - US cell phone providers (pairwise roaming agreements)
  - IE HTML, Netscape HTML, etc.
  - WAP (WML same purpose as HTML, but not compatible)
- network value = $O(n^2)$, (Metcalfe's Law)
- pairwise translation: cost = $O(n^2)$, utility = $O(1)$
- CIR: cost = $O(n)$, utility = $O(n)$

Challenge 1: Different Address Formats

- Options:
  - Map one address format to another. Why not?
  - Provide one common format
    - map lower level addresses to common format
- Format:
  - Initially: 8b network 16b host 24b total
  - Before Classless InterDomain Routing (CIDR):
    - 7b/24b, 14b/16b, or 21b/8b 32b total
  - After CIDR: Arbitrary division 32b total
  - NAT: 32b + 16b simultaneously active
  - IPv6: 128b total

Address Formats

- 256 networks? What were they thinking?
- Why CIDR?
- What happens if they run out before IPv6?
- Why IPv6?
- Why 128b for IPv6? 2^{48}=281 trillion.
- Why not variable length addresses?

Challenge 2: Different Packet Sizes

- Need to define maximum packet size
- Options:
  - Take the minimum of the maximum packets sizes over all networks
  - Implement fragmentation/reassembly
    - Flexibility to adjust packet sizes as new technologies arrive
    - IP: fragment at routers, reassemble at host
    - Why reassemble at routers?
  - Still stuck with 1500B as de facto maximum

Other Challenges

- Errors → require end-to-end reliability
  - Thought to be rarely invoked, but necessary
- Different (routing) protocols → coordinate these protocols
- Accounting
  - Did not envision script kiddies
- Quality of Service
  - Not addressed

Transmission Control Program

- Original TCP/IP (Cerf & Kahn)
  - no separation between transport (TCP) and network (IP) layers
  - one common header (vestige?)
  - flow control, but not congestion control (why not?)
  - fragmentation handled by TCP
- 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

Devil's Advocate

- Who cares about resource sharing?
  - 1974: cycles, storage, bandwidth expensive, people cheap
  - 2002: resources cheap, people expensive
  - 1974: Share computer resources
  - 2002: Communicate with people, access documents, buy, sell
- Does it still make sense to make processes the endpoint?
- Where do people and organizations fit into the ISO layering model?

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)
  - failures (excepting network partition) should be transparent to endpoints
- maintain state only at end-points (fate-sharing)
  - no need to replicate and restore router state
  - disadvantages?
- Internet: stateless network architecture
  - no per-flow state, still have state in address allocation, DNS

2. Types of Services

- Add UDP to TCP to better support other types of applications
  - e.g., “real-time” applications
- Probably main reason for separating TCP and IP
- Provide datagram abstraction: lower common denominator on which other services can be built
  - service differentiation considered (ToS header bits)
  - was not widely deployed (why?)

Application Assumptions

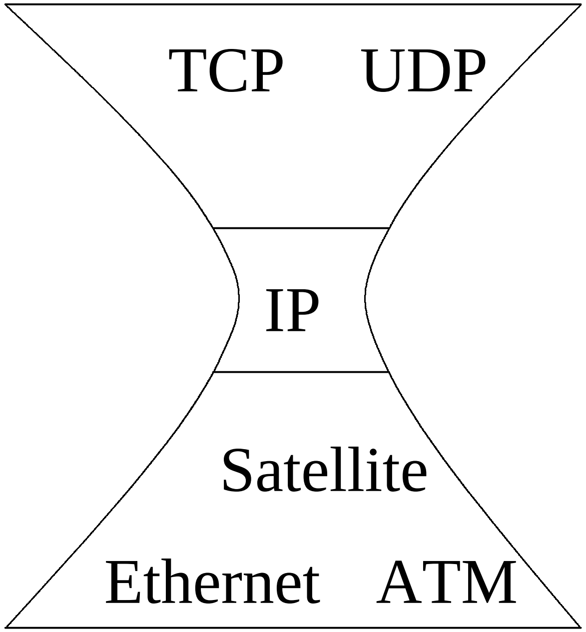
- Who made them:
  - Telephone network: voice (web, video?)
  - Cable: broadcast (2-way?)
  - X.25: remote terminal access (file transfer?)
  - BBS: centralized meeting place (web, p2p?)
  - NAT: client/server model (p2p, IM, IP Telephony?)
- Who didn't: Internet
  - Caveat: best-effort, unicast, fixed location (real-time, multicast, mobility?)
- Allows development of unforseen applications:
  - Web, p2p, distributed gaming
- Sometimes too general:
  - Interdomain, multi-source multicast scales poorly
  - Single source multicast scales better [Holbrook99]

3. Variety of Networks

- Very successful
  - 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, single delivery, QoS guarantees
- The mantra: IP over everything
  - Then: ARPANET, X.25, DARPA satellite network..
  - Now: ATM, SONET, WDM, PPP, USB, 802.11b, GSM, GPRS, DSL, cable modems, power lines

Internet Architecture

- Packet-switched datagram network
- IP is the glue
- Hourglass architecture
  - all hosts and routers run IP
- Common Intermediate Representation



Other Goals

- Allow distributed management
  - each network can be managed by a different organization
  - different organizations need to interact only at the boundaries
  - doesn't work so well for routing, accounting
- Cost effective
  - sources of inefficiency
    - header overhead
    - retransmissions
    - routing
  - …but routers relatively simple 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)
    - Moore’s law made this moot point, both <$100
  - bad implementations or malicious users can produce considerably harm (remember fate-sharing?)
    - DDoS possibly biggest threat to Internet
- 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: Minimalist Approach

- Dumb network
  - IP provide minimal functionalities to support connectivity
  - addressing, forwarding, routing
- Smart end system
  - transport layer or app does more sophisticated functionalities
  - flow control, error control, congestion control
- Advantages
  - accommodate heterogeneous technologies
  - support diverse applications (telnet, ftp, Web, X windows)
  - decentralized network administration
- Disadvantages
  - poor realtime performance
  - poor accountability