Internet Architecture and Assumptions

David Andersen
CMU Computer Science

Waitlist

- ~14 slots in the class
- ~11-13 students enrolled
- 16 on waitlist
  - 2 Ph.D.
  - 14 MS
- Outlook not *too* good for the MS students.
- May admit one or two.
- Priority to those who attend class

Course goals

- Examine scenarios that make traditional networking difficult, and various techniques that can / have / should be used to cope with those challenges.
- How?
  - Lecturelets: Snippets of background
  - Readings: A mix of classical and cutting-edge research
  - Semester-long research project

Syllabus, etc.

- Syllabus details…
- Watch the web page! Later lectures still evolving, and we'll adjust based on the course progress.
- Background books
  - Peterson (general) - several on reserve
  - Stallings (wireless) - two on reserve
  - Let me know if we need more.
- Office hours TBA. We'll vote on time slots after roster stabalizes a bit.
  - For now, email for appointment.

Grading

- 30% discussion leading (3x 10%)
- 10% class participation / attendance
- 60% project

- This is a grad seminar. I expect to give all A's (… and expect attendance/participation/good projects).

Discussion leading

- Each lecture:
  - dga supplies background
  - One group of two students presents paper summary / prepares discussion questions
- 24 lectures. ~14 students.
- Each group responsible for 3 lectures
- Signups next Monday (9/19).
  - Think about the topics you might want to present / glance at papers on syllabus
  - Next mon: Must cover next 7 lectures
  - VOTE: Assigned topics or first come first served?

Projects

- Semester-long research project
- Must be topical. Fairly wide interpretation - availability, reliability, wireless, ad hoc, mobility, sensor nets, …
- Semi-novel. SIGCOMM quality not required, but goal should be a project that could be a conference paper with some more work.
- Proposals due 10/12. *I will happily review and provide feedback earlier!*

Project deliverables 1

- Proposal:
  - 1 page proposed research summary
  - Meeting to discuss plans
- Review:
  - Meeting with instructor to discuss progress, bottlenecks, etc.
- Presentation (12/05 and 12/07)
  - 20 minute (ish) conference-style talk about the research
- Paper (due 12/07)
  - 5-10 page conference-style writeup

Internet Architecture

- Background
  - "The Design Philosophy of the DARPA Internet Protocols" (David Clark, 1988).
- Fundamental goal: Effective network interconnection
- 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

Priorities

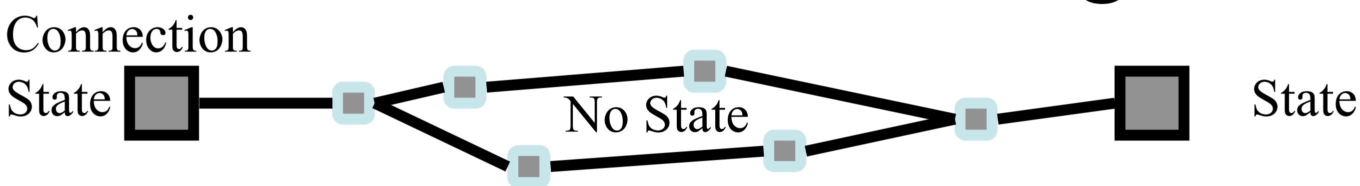
- The effects of the order of items in that list are still felt today
  - E.g., resource accounting is a hard, current research topic
- Let’s look at them in detail

Survivability (more later)

- If network disrupted and reconfigured
  - Communicating entities should not care!
  - No higher-level state reconfiguration
  - Ergo, transport interface only knows “working” and “not working.” Not working == complete partition.
- How to achieve such reliability?
  - Where can communication state be stored?

	Network	Host
Failure handing	Replication	“Fate sharing”
Net Engineering	Tough	Simple
Switches	Maintain state	Stateless
Host trust	Less	More

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
- Survivability compromise: Heterogenous network -> less information available to end hosts and Internet level recovery mechanisms

Types of Service

- TCP vs. UDP
  - 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
    - Made Internet datagram service less useful!
  - Hard to implement without network support
  - QoS is an ongoing debate…

Varieties of Networks

- Discussed a lot of this last time -
  - Interconnect the ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links…
- Mininum 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

The “Other” goals

- Management
  - Today’s Internet is decentralized - BGP
  - Very coarse tools. Still in the “assembly language” stage
- Cost effectiveness
  - Economies of scale won out
  - Internet cheaper than most dedicated networks
  - Packet overhead less important by the year
- Attaching a host
  - Not awful; DHCP and related autoconfiguration technologies helping. A ways to go, but the path is there

Accountability

- Huge problem.
- Accounting
  - Billing? (mostly flat-rate. But phones are moving that way too - people like it!)
  - Inter-provider payments
    - Hornet's nest. Complicated. Political. Hard.
- Accountability and security
  - Huge problem.
  - Worms, viruses, etc.
    - Partly a host problem. But hosts very trusted.
  - Authentication
    - Purely optional. Many philosophical issues of privacy vs. security.

Challenging Environments

- Focus: How do these environments challenge the assumptions behind the Internet architecture?

Challenging Environments

- Wireless
- Host mobility
- Ad hoc wireless networks
- Satellite
- Space
- Sensor networks
- Dial-up / store and forward
- Disconnection
- High availability requirements

Wireless

- Burst losses / fading / multipath / interference
  - Microwave ovens
  - Big, mobile microwave-absorbing barriers (us)
  - Weather, etc.

- “Reasonable” packet delivery odds?
- 0-90% packet loss common

Mobility

- Not really considered in original arch.
- Changing IP addresses
  - Breaks TCP connections
  - Fundamental problem: Identity vs. topology
    - IP address is a topological identifier, *not* a user or host identifier
- Temporary disconnection during movement
  - Applications often don't know how to cope

Ad Hoc wireless

- Create a network from an extant collection of wireless nodes
  - Run a routing protocol between them
- All the problems of wireless, plus:
- Unprovisioned
  - Nobody planned the links, nodes, etc.
- Dynamic
  - Nodes/links come and go *much* more frequently than they do on the wired Internet

Satellite

- Lossy, like other wireless
- High delay: 100s of ms.
  - Often high delay * bandwidth product
- Long term disruption (satellite goes out of view, etc.)

Space

- What's the round-trip delay to Mars?
  - 6.5 minutes (best), 44 minutes (worst).
- *Totally* shatters the assumptions behind many Internet protocols (TCP) and applications that assume timeouts of seconds.
- Occlusion: Planets rotate, get in each others' way, etc.
  - Challenge: How do you route a message from here to mars?

Sensor networks

- Deployment of small, usually wireless sensor nodes.
  - Collect data, stream to central site
  - Maybe have actuators
- Hugely resource constrained
  - Internet protocols have implicit assumptions about node capabilities
  - Power cost to transmit each bit is very high relative to node battery lifetime
  - Loss / etc., like other wireless
  - Ad-hoc: Deployment is often somewhat random

Disconnection / store & forward

- Many Internet protocols assume frequent connectivity
- What if your node is only on the Internet for 5 minutes every 6 hours?
  - How do you browse the web?
  - Receive SMTP-based email?

High availbility requirements

- No QoS assumed from below
- Reasonable but non-zero loss rates
  - What's minimum recovery time?
    - 1rtt
  - But conservative assumptions end-to-end
    - TCP RTO - min(1s)!
- Interconnect independent networks
  - Federation makes things hard:
    - My network is good. Is yours? Is the one in the middle?
  - Scale
    - Routing convergence times, etc.

End-to-end Arguments Arguments

- What functions can only be implemented correctly with the help of the endpoints?
  - Challenging environments expose problems that require more endpoint support, e.g., end-to-end reliable delivery.

- What functions can *not* be implemented without the help of the network?
  - Challenging environments start to expose more of these functions, too. E.g., retries over wireless.