Advanced Computer Networks

Internet Design

Jianping Pan
Summer 2007

Internet architecture and protocols

- [CK74] V. G. Cerf and R. E. Kahn, "A Protocol for Packet Network Interconnection". IEEE Transaction on Communications, 22(5), May 1974, pp. 637-648. [TCPdesign]
- [SRC84] J. Saltzer, D. Reed, and D. Clark, "End-to-end Arguments in System Design". ACM Transactions on Computer Systems, Vol. 2, No. 4, 1984, pp. 195-206. [end2end]
- [Cla88] D. Clark, "The Design Philosophy of the DARPA Internet Protocols". In Proceedings of ACM SIGCOMM '88, 106-114, Palo Alto, CA, Sept 1988. [IPSdesign]

[Cla88] The design of IPS

- Originally published in Proc SIGCOMM'88
- A *retrospective* “design” document
  - “TCP/IP, was first proposed fifteen years ago.”
  - “it is sometimes difficult to determine the motivation and reasoning which led to the design.”
  - “In fact, the design philosophy has evolved considerably from the first proposal [CK74] to the current standards.”
    - “datagram does not receive particular emphasis”
    - “layering the architecture into the IP and TCP layers”

Fundamental goal

- “The top level goal for the DARPA Internet Architecture was to develop an *effective* technique for *multiplexed* utilization of existing *interconnected* networks.”
  - interconnection vs integration
  - packet switching vs circuit switching
  - store-and-forward packet switching
    - the implication
- Goals scoreboard

Second level goals

- “in order of importance”
  - “continue despite loss of networks or gateways.”
  - “support multiple types of communications service.”
  - “accommodate a variety of networks.”
  - “permit distributed management of its resources.”
  - “cost effective.”
  - “permit host attachment with a low level of effort.”
  - “accountable.”

Survivability

- DARPA-funded
- “In other words, at the top of transport, there is only one failure, and it is total partition.”
  - layering transparency
  - end-to-end vs hop-by-hop
    - fate-sharing
  - stateless vs stateful switches
    - dumb networks
  - middle-box?!

Types of services

- Network is essentially driven by application requirements and communication technologies
  - QoS: throughput, delay (jitter), loss
  - e.g., rlogin, ftp, email, VoIP, IPTV, MMOG
- The separation of TCP and IP
  - TCP: reliable, stream-like
  - UDP: unreliable, datagram
  - reliable “network”?

Varieties of networks

- IP sits on a minimum set of assumptions
  - move a packet
  - of a reasonable minimum packet size
  - with a reasonable delivery success ratio
  - addressing capability, if not point-to-point link
- Not assumed
  - reliable delivery, broadcast/multicast, priority queuing, internal knowledge, etc
  - very “heterogeneous” networks?

Distributed management

- Internet: a network of networks
  - autonomous systems (AS)
  - tiers of service providers
  - hierarchical naming
  - hierarchical addressing*
  - hierarchical routing
  - distributed “coordination”?

Cost effectiveness

- Multiplexing gain
  - store-and-forward packet switching
- Layered architectures
  - similar functions in different layers
- Packet headers
  - packet header vs user payload
- Protocol mechanisms
  - end-to-end vs local retransmission

Easy attachment

- Requirements on end systems
  - anything says TCP/IP
  - smart hosts vs smart networks
- Requirements on intermediate systems
  - anything says IP and knows routing
    - IP: one number, two roles
    - i.e., addressing, and
    - routing
- IP mobility

Network accountability

- Application level
  - email spam?
- Session/call level
- Flow/connection level
- Packet level
  - spoofed source IP address?
- Security?
  - authentication, authorization, accounting
  - confidentiality, integrity, authenticity, availability

Goals scoreboard

- Well achieved
  - list:
- Not so well achieved
  - list:
- Not achieved
  - list:
- Other goals
  - should be considered then
  - should be considered now

New networking environments

- Wireless networks
  - infrastructure-based or infrastructure-less
- Very high-speed networks
- Very “long” networks
  - interplanetary
- Resource very constrained networks
  - micro-sensor networks: power, computation, storage, communication
- Often “disconnected” networks

New application requirements

- Quality of service
  - throughput, delay (jitter), loss
- High availability
- High scalability
- Security
- And more...

[SRC84] End-to-end arguments

- One of the design principles of the Internet
- “Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer.”
- “Design principles that provide guidance in this choice of function placement are among the most important tools of a system designer”
- “Discusses one class of function placement that has been used for many years with neither explicit recognition nor much conviction”

A typical setting

- A networked computer system
  - communication subsystem (intermediate system)
  - the rest of the system (end system)
- A list of functions to be implemented
  - in intermediate system?
  - in end system?
  - in both intermediate and end system?
    - in collaboration, or
    - in redundancy

An example

- Careful file transfer
  - from computer A to B
  - across the communication subsystem
- Where can things go wrong?
  - almost every where
  - read error at A
  - process error at A
  - communication error
  - process error at B
  - write error at B

Possible approaches

- For each step
  - duplication
  - timeout and retry
  - error detection and recovery
  - crash recovery
  - goal: reduce error probability everywhere
- For end-to-end
  - checksum generated at A
  - checksum verified at B
  - if checksum fails, end-to-end retransmission

The end-to-end argument

- “The function in question can completely and correctly be implemented only with the knowledge of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible.”
- The messages
  - do it only when you can do it best
  - do it only where it really matters

Performance aspects

- Some lower level functions are helpful
  - e.g., non-persistent local error recovery
  - no need to provide perfect reliability
  - still cannot replace higher level functions
    - i.e., for performance, not for correctness
  - some can be replaced by higher level functions
    - e.g., multi-block vs file checksum
  - some may not be needed by all applications

More examples

- Delivery acknowledgment
  - e.g., delivery notice to end-host or endpoint
  - acknowledgment piggyback
- Secure data transmission
  - e.g., end-to-end encryption
  - encryption keys and parameters
- Duplicate message suppression
- FIFO message delivery
- Transaction management

Where is the “end”?

- Application specific
  - e.g., conversation vs playback
- The application of “end-to-end argument”
  - Internet architecture
    - dumb networks
  - TCP end-to-end control
    - flow, error, congestion
- The end of “end-to-end” arguments?
  - middle-box
  - cross-layer design

This lecture

- Internet design
  - at architecture level
  - design goals
  - end-to-end arguments
- Do not forget A0 (due Friday, May 11 by email)
  - set Introduction slides
- Explore further
  - [CT90] D. Clark and D. Tennenhouse, "Architectural Consideration for a New Generation of Protocols". In Proceedings of ACM SIGCOMM '90, Philadelphia, PA, September 1990. [ALF/ILP]

Next lecture

- The evolution of the Internet architecture
  - required reading
    - [She95] S. Shenker, "Fundamental Design Issues for the Future Internet". IEEE Journal on Selected Areas in Communications, Vol. 13, No. 7, September 1995, p p. 1176-1188.
    - [FG01] P. Francis and R. Gummadi. "IPNL: A NAT-extended Internet architecture." In Proceedings of ACM SIGCOMM, San Diego, CA, Aug. 2001. [IPNL]
    - [CWRB02] D. Clark, J. Wroclawski, K. Sollins, and R. Braden, Tussle in Cyberspace: Defining Tomorrow's Internet , Proceedings of ACM SIGCOMM '2002. [tussle]
    - [SAZSS04] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana, "Internet indirection infrastructure," IEEE/ACM Trans. Networking, Vol. 12, No. 2, pp. 205- 218. [I3]

More on the course

- Course presentation
  - pick topics from the reading list
    - Internet design, network architectures
    - overlay networks, peer-to-peer networking
    - congestion control
    - network routing
    - traffic management
    - network characterization
  - choose papers from the reading list
    - the reading list is still being updated
    - you can recommend papers (not in the list yet)!

More on the course

- Course project
  - any topic related to computer networks
    - it's your job to justify (and I can help)
  - proof-of-concept prototype
    - possible approaches: analysis, simulation, emulation, experimentation, implementation, measurement
      - pick at least two of the above approaches
  - final deliverables
    - project report
    - project presentation

Other systems courses at UVic CS

- Computer Networks (CSc 450)
- Embedded systems (CSc 460)
- Multimedia systems (CSc 461)
- Distributed systems (CSc 462)
- Fault-tolerant (CSc 454) and Parallel (464)
- Topics in systems (CSc 485A-H)
  - advanced operating systems
  - advanced communication networks
    - focus on layer 2 and 3
  - wireless mobile networks, etc

Course policies

- See official course outline
  - late assignments, mark appeals, etc
  - academic integrity
  - accommodation, etc
- Summaries, presentations, project
  - collaboration/participation is encouraged
  - responsibility: your submitted work is yours
    - for undergraduate group project, you need to identify individual's contribution clearly in documentation
  - obligation: give credits to references

Guest lecture

- Guest lecture for csc450/550
  - by Ron Kozsan, UVic Network Services Manager
  - on “UVicNet, BCNET, CA*Net4, and more”
  - May 10, 11:30am-1pm, MACD116
- You are very welcome to attend
  - come to find out “what's under the hood” for the things you use most!