Midterm Review

CSE 486/586: Distributed Systems

Ethan Blanton

Department of Computer Science and Engineering

University at Buffalo

The Internet (pt. 1)

- The Internet is a network of networks
- IP will run over many networks because it makes few assumptions
- IP provides very limited service
- Transport protocols ride on top of IP
- UDP is connectionless datagrams
- TCP is connected byte streams

The Internet (pt. 2)

- The end-to-end argument provides guidance on where to implement functionality.
- TCP provides services that IP does not.
- The TCP model is an full duplex in-order byte stream.
- TCP loss recovery is effected by a distributed state machine.

The Go Language

- Go is unique, meet it on its own terms
- Go is a picky language
- Idioms are worth learning
- Go uses structural (“duck”) typing
- Go provides polymorphism through
  - Methods
  - Interfaces

A Model of Distributed Systems

- Distributed systems communicate by message passing
- We will work with asynchronous systems
- Delay is indistinguishable from loss
- Concurrent execution can lead to races
- Happens before is the cure for races
- CSP is a programming model for message passing

Failure and Failure Detection

- Failure detection is important for distributed systems.
- There are many possible types of failures:
  - Crash
  - Omission
  - Response
  - Byzantine
- Crash failures and message loss can be confused.
- Completeness and accuracy are measures of failure detector goodness.
- Asynchronous system failure detectors cannot be both.

Time

- Time is important to distributed systems
- There are standards for measuring time
- Different clock technologies have strengths and weaknesses
- Clocks experience relative phase and frequency errors
- Synchronization protocols must deal with network delays
- NTP provides robust synchronization over Internet paths

Logical Time

- Logical clocks track causality of events
- Lamport clocks use a single integer to define causality
- Vector clocks provide greater precision than Lamport clocks, but require more state
- Logical clock orderings can be partial or total

Global States

- Global states are useful for many purposes
- A consistent global state could have happened
- Consistency is ensured by preserving happens before
- Chandy-Lamport snapshots capture global state
  - More work is needed without reliable, ordered messages

Naming in Distributed Systems

- Naming is hard, and harder for distributed systems
- Naming can be:
  - Centralized at some authority
  - Delegated hierarchically
  - Distributed via global uniqueness
- DNS is a global distributed database that:
  - Delegates authority
  - Provides redundancy
  - Uses caching to improve performance

Distributed Hash Tables

- Distributed hash tables use globally unique names to avoid naming authorities
- Names are often cryptographically secure hash values
- DHTs provide key-value lookups in $O(\log n)$ messages, where n is the size of the key space
- Kademlia is a DHT with desirable properties:
  - Robust to adverserial nodes
  - Deals well with churn
  - Self-maintaining structure
- Kademlia is used in large distributed systems

Broadcast and Multicast

- Distributed systems benefit from group communication
- Internet communication is mostly unicast
- Broadcast and multicast can be built from unicast
- Relatively simple protocols can achieve all-or-nothing delivery
- FIFO delivery requires only a TCP-like sequence number

Ordered Multicast

- Safety means constraints will never be violated
- Liveness means every message is eventually delivered
- ISIS provides causally and totally ordered multicast
- The VT protocol uses vector clocks to causally order
- ISIS ABCAST uses distributed sequencing to totally order

Gossip Protocols

- Gossip protocols provide probabilistic delivery
- Cost is usually about $c \cdot |G| \log |G|$ per message
- Lightweight Probabilistic Broadcast solves:
  - Changing group membership
  - Process membership knowledge overhead for very large $|G|$

Leader Election

- Centralized authority doesn't mean permanent authority
- Distributed elections can be held
  - Bully algorithm
  - Ring algorithm
- Global identifiers keep cropping up
- Proof of work can make global IDs safer
- Security guarantees require threat models

DARPA Protocol Design

The Design Philosophy of the DARPA Internet Protocols [3]

- Fundamental goal
- Seven second-level goals
- How have these goals led to the Internet 50 years later?

The End-to-End Argument

End-to-end Arguments in System Design [8]

- What do end-to-end protocols achieve?
- When should protocols be end-to-end?
- Do we see end-to-end protocols in the distributed protocols we've looked at?
  - Why?
  - Why not?

Communicating Sequential Processes

Communicating Sequential Processes [5]

- How does the CSP model tame concurrency?
- How is CSP different from Go channels?
- Things to understand:
  - Functions
  - Coroutines (these are always hard!)

Unreliable Failure Detectors

Unreliable failure detectors for reliable distributed systems (Preliminary Version) [1]

(Only through Section 3.)

- Completeness
- Accuracy
- Why is this important for asynchronous systems?
- How can the other protocols we've looked at use failure detection?

Logical Clocks

Time, Clocks, and the Ordering of Events in a Distributed System [6]

- When are logical clocks appropriate?
- How do Lamport clocks compare to vector or other clocks?
- Where do we see logical clocks in other protocols?

Kademlia

Kademlia: A Peer-to-peer Information System based on the XOR Metric [7]

- Why is searching efficient?
- How does it achieve reliability from failed nodes?
- Why is the XOR metric “unidirectional”?

Gossip

Lightweight Probabilistic Broadcast [4]

- Handles group membership and messaging, both.
- Why might you use gossip?
- Can gossip replace other communication paradigms?
- What are the implications of partitioning?

Ring Elections

An Improved Algorithm for Decentralized Extrema-finding in Circular Configurations of Processes [2]

- Why is the title longer than the paper?
- How does this compare to bully elections?
- How do deadlock and node failure relate?

References I

Required Readings

[1] Tushar Chandra and Sam Toueg. "Unreliable failure detectors for reliable distributed systems (Preliminary Version)". In: July 1991, pp. 325–340. URL: https://dl-acm-org.gate.lib.buffalo.edu/doi/10.1145/112600.112627.

[2] Ernest Chang and Rosemary Roberts. "An Improved Algorithm for Decentralized Extrema-finding in Circular Configurations of Processes". In: 22.5 (May 1979), pp. 281–283. DOI: 10.1145/359104.359108. URL: https://search.lib.buffalo.edu/permalink/01SUNY_BUF/12pkqkt/cdi_crossref_primary_10_1145_359104_359108.

References II

[3] David D. Clark. “The Design Philosophy of the DARPA Internet Protocols”. In: *Computer Communication Review* 18.4 (Aug. 1988), pp. 106–114. URL: http://ccr.sigcomm.org/archive/1995/jan95/ccr-9501-clark.pdf.

[4] Patrick T. Eugster et al. “Lightweight Probabilistic Broadcast”. In: *Proceedings of the IEEE International Conference on Dependable Systems and Networks*. IEEE, July 2001, pp. 443–452. DOI: 10.1109/dsn.2001.941428. URL: http://se.inf.ethz.ch/people/eugster/papers/lpbcast.pdf.

References III

[5] C. A. R. Hoare. “Communicating Sequential Processes”. In: 21.8 (Aug. 1978), pp. 666–677. URL: https://search.lib.buffalo.edu/permalink/01SUNY_BUF/12pkqkt/cdi_crossref_primary_10_1145_359576_359585.

[6] Leslie Lamport. “Time, Clocks, and the Ordering of Events in a Distributed System”. In: 21.7 (July 1978). Ed. by R. Stockton Gaines, pp. 558–565. URL: https://dl-acm-org.gate.lib.buffalo.edu/doi/pdf/10.1145/359545.359563.

References IV

[7] Petar Maymounkov and David Mazières. "Kademlia: A Peer-to-peer Information System based on the XOR Metric". In: *Proceedings of the International Workshop on Peer-to-Peer Systems*. Mar. 2002, pp. 53–65. URL: https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf.

[8] Jerome H. Saltzer, David P. Reed, and David D. Clark. "End-to-end Arguments in System Design". In: 2.4 (Nov. 1984), pp. 277–288. URL: http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf.