

CPSC 416 – Distributed Systems

Lecture 18: The Impossibility Zoo

UBC Vancouver

2026-03-12

From Last Time

Turn to your neighbor. 2 minutes.

The Byzantine bound says we need $3f + 1$ nodes to tolerate f liars.

Why $3f + 1$ and not $2f + 1$? What breaks at $3f$?

(Pairs: 2 minutes)

Failure-First Question

We've seen FLP and the Byzantine bound. We've seen CAP and the Two Generals problem.

Are these separate results, or instances of something deeper?

Today we look at the zoo of impossibility results—and find that **every cage has the same lock.**

Four Impossibility Results

You already know all of these. Let's line them up.

- ① **FLP** (1985): No deterministic consensus in async with one crash
- ② **CAP** (2000/2012): Can't have Consistency + Availability + Partition tolerance
- ③ **Two Generals** (1975): Can't reach certain agreement over an unreliable channel
- ④ **Byzantine bound** (1982): Can't agree with $< 3f + 1$ nodes when f lie

Four results, four decades, four different communities.

Same structure.

FLP: The Adversary Who Only Delays

You saw this in L12.

- Proposer A completes Phase 1. Sends $\text{ACCEPT}(1, v)$.
- The network delays A's message by one round-trip.
- Proposer B's $\text{PREPARE}(2)$ arrives first. Quorum re-promises.
- A's ACCEPT arrives late. Rejected.
- Repeat forever.

Freedom: Asynchrony (messages take arbitrary time).

Lost property: Liveness (eventually decide).

The adversary never lies. It just delays.

CAP: The ATM During a Partition

Scenario: Your bank has two data centers. The link between them goes down.

A customer walks into the Oregon branch. Balance: \$500.
They want to withdraw \$400.

Option A: Allow the withdrawal.

- **Available**—the customer gets their money.
- But someone in Virginia might *also* withdraw \$400.
- **Consistency violated**. Bank loses \$300.

Option B: Refuse the withdrawal.

- **Consistent**—no risk of double-spend.
- But the customer can't use their account.
- **Availability violated**. Customer leaves angry.

CAP: The Structure

Freedom: Network partitions can occur (and you can't prevent them).

Lost property: Consistency + Availability (pick at most one during a partition).

Notice: When the partition heals, you can have both again.

The impossibility is *during* the partition, not forever.

This is why CAP is about trade-offs, not absolute limits.

Two Generals: The Unreliable Messenger

Scenario: Two armies on opposite sides of a valley. They must attack simultaneously or they'll lose.

General A sends a messenger: "Attack at dawn."

The messenger might be captured. A doesn't know if B got the message.

B sends an acknowledgment. The ack might be captured.

A sends an ack-of-ack. It might be captured.

...

Freedom: Any message can be lost.

Lost property: Certain mutual agreement (both know the other will act).

No finite number of messages suffices.

Byzantine: The 3-Node Counting Argument

You saw this in L17.

Three nodes: A , B , C . One is Byzantine ($f = 1$).

- A proposes 0. C proposes 1. B is the traitor.
- B tells A : "I agree with 0."
- B tells C : "I agree with 1."
- A sees two votes for 0, one for 1. Decides 0.
- C sees two votes for 1, one for 0. Decides 1.
- **Disagreement. Safety violated.**

Freedom: f nodes can lie arbitrarily.

Lost property: Agreement with $< 3f + 1$ nodes.

Checkpoint 1

Articulate:

Each result takes something away.

In your own words: what is the “something” each result identifies as the source of the problem?

(Pairs: 3 minutes)

The Common Structure

Every impossibility result has the same skeleton:

Given freedom F , property P is unachievable.

The **freedom** is what the environment can do to you.

The **property** is what you wanted to guarantee.

The result says: The environment has enough latitude to prevent you from verifying P from inside the system that's trying to produce P .

The Impossibility Zoo

Result	Freedom	Lost Property	Escape
FLP (1985)	Asynchrony	Liveness	Partial synchrony, randomization
CAP (2000)	Partitions	Consistency + Availability	Weaken one during partition
Two Generals (1975)	Unreliable channel	Certain agreement	Probability, reliable sub-channel
Byzantine (1982)	f liars	Agreement with $< 3f+1$ nodes	Signatures, more nodes, trusted HW

The Structural Insight

You cannot verify X from inside a system
that has enough freedom to prevent verification.

- FLP: You can't verify "all processes decided" when you can't tell slow from crashed.
- CAP: You can't verify "this is the latest value" when you can't reach the other partition.
- Two Generals: You can't verify "they got my message" when the channel eats messages.
- Byzantine: You can't verify "this node is honest" when f nodes can lie about anything.

Every impossibility is a verification impossibility.

Checkpoint 2

Articulate:

If every impossibility result is really about *verification*, then what should every *escape* have in common?

(Pairs: 2 minutes)

Every Escape Works the Same Way

Two moves. That's it.

Move 1: Restrict the freedom.

- FLP → Partial synchrony: “Messages eventually arrive within some bound.”
- CAP → Weaken C or A: “We accept stale reads during partitions.”
- Two Generals → Reliable sub-channel: “This link doesn't lose messages.”
- Byzantine → Signatures: “Liars can't forge honest nodes' messages.”

Every Escape Works the Same Way (cont.)

Move 2: Add something external.

- FLP $\rightarrow$ Randomization: An external source of non-determinism.
- CAP $\rightarrow$ Conflict resolution: External merge logic (CRDTs, last-writer-wins).
- Two Generals $\rightarrow$ Probability: “If I send k messengers, the chance all are captured is $(1 - p)^k$.”
- Byzantine $\rightarrow$ Trusted hardware: A component the adversary can't subvert.

Restrict the freedom, or bring in something the system can't produce on its own.

The Escape Pattern

Impossibility: Freedom F blocks property P

↙

Restrict F

Partial synchrony

Signatures

Reliable sub-channels

Weaken the environment

↘

Add External

Randomness

Trusted hardware

Merge logic (CRDTs)

Strengthen the system

↙ ↘

Property P recoverable

under the new assumptions

Testing the Framework: Flexible Paxos

From L14: Flexible Paxos (Howard 2016) showed that Phase 1 and Phase 2 quorums don't need to be the same size.

Is this an impossibility escape?

- The “impossibility”: Majority quorums seem required for Paxos.
- The “freedom”: Our assumption that both phases need the same quorum.
- The escape: **Restrict the assumption**. Only cross-phase intersection is needed.

Not a formal impossibility result—but the same pattern.

A false “impossibility” imposed by convention, not by proof.

Howard didn't break the proof. She read it more carefully than everyone else.

Testing the Framework: Spanner

Google Spanner claims to be “effectively CA” (consistent and available) despite CAP.

How?

- **Restrict the freedom:** Google controls its own network. Partitions between Spanner nodes are *extremely* rare.
- **Add something external:** TrueTime (GPS + atomic clocks) provides bounded uncertainty on real time.
- **Trade-off:** Commit-wait. Transactions stall for the uncertainty window (typically $< 7\text{ms}$).

Spanner doesn't violate CAP. It makes partitions so rare that the C+A trade-off almost never triggers.

We'll study this in detail in L19.

Checkpoint 3

Articulate:

Spanner uses TrueTime (GPS + atomic clocks) to bound clock uncertainty.

In our framework: is TrueTime “restricting the freedom” or “adding something external”?

Could it be both?

(Pairs: 2 minutes)

Activity: Analyze a New Claim

Claim:

“No replicated database can provide both strong consistency and sub-millisecond latency at global scale.”

Your task:

- 1 Is this an impossibility result with the same structure?
- 2 What is the **freedom**?
- 3 What is the **lost property**?
- 4 What are the possible **escapes**?

(Groups of 3: 8 minutes. Then present.)

Activity Debrief

Freedom: The speed of light. Messages between continents take $\geq 30\text{ms}$ round-trip. Physics, not engineering.

Lost property: Strong consistency (linearizability) requires coordination. Coordination requires round-trips. Round-trips at global scale exceed 1ms.

Escapes:

- **Restrict freedom:** Don't go global. Keep replicas close. (Defeats the purpose.)
- **Weaken the property:** Causal consistency, eventual consistency, session guarantees.
- **Add external:** Synchronized clocks (Spanner's commit-wait) reduce but don't eliminate latency.
- **Hybrid:** Strong consistency for some operations, weak for others (e.g., Spanner vs. Dynamo).

Same skeleton. The freedom here is physics, not protocol design.

Bridge: Why This Pattern Keeps Appearing

[Optional section — skip if time is tight]

The impossibility pattern is not limited to distributed systems.

Consider: A model evaluating its own output.

- The model generates text.
- The model checks: “Is this text correct?”
- But the checker is the *same system* that produced the text.

Freedom: The model can produce any output, including outputs that look correct but aren’t.

Lost property: Reliable self-verification.

A model evaluating its own output is Paxos trying to guarantee its own liveness.

Same structure, same escape: an *external* observer, telemetry the system doesn’t control.

Bridge: The Escape is Always External

[Optional section — skip if time is tight]

In distributed systems: the escape is partial synchrony, trusted hardware, an external randomness source.

In self-evaluation: the escape is an external evaluator, ground-truth data, a test the model didn't write.

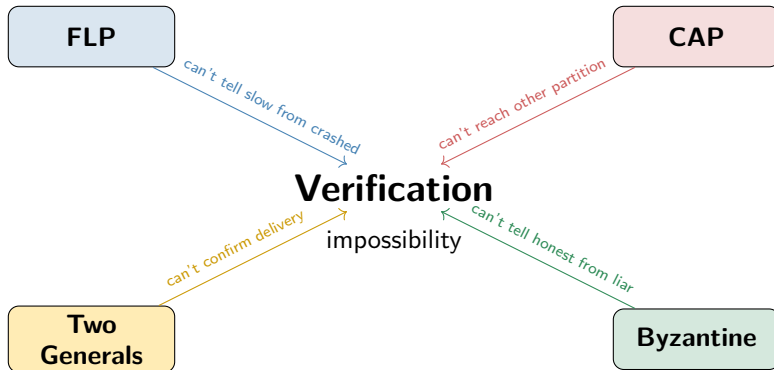
The structural insight:

You cannot verify X from inside the system that produces X ,
when the system has enough freedom to defeat verification.

This is not a metaphor. It is the same formal structure.

We'll leave this here. The distributed systems version is what's on the exam.

The Zoo, Organized



Same lock. Same two keys.

Evidence Question

You're designing a new distributed protocol.

Someone claims it achieves strong consistency, always terminates, and tolerates arbitrary asynchrony.

Which impossibility result tells you this claim is false?

What question would you ask to find the hidden assumption?

(This is the skill. The results are tools for evaluating claims.)

What We Learned

- Every impossibility result says: **given freedom F , property P is unachievable**
- The core mechanism: the system has enough freedom to prevent internal verification of P
- **FLP**: asynchrony blocks liveness verification
- **CAP**: partitions block consistency verification
- **Two Generals**: unreliable channels block agreement verification
- **Byzantine**: liars block identity verification
- Every escape either **restricts the freedom** or **adds something external**

The practical payoff:

When someone claims to “solve” an impossibility, ask: *What assumption did you add? What did you give up?*

Next: L19—Spanner’s escape from CAP.

- **No reading assigned** (this is a synthesis lecture)
- **Next lecture (L19):** Geo-Replication and Spanner—CAP escape in practice
- **Reading for L19:** Spanner (Corbett et al., 2012)
- Self-assessment opens 13:00, due next class 11:30 (Pacific)
- Capstone milestone check-in: see Canvas

Use the framework. When you encounter a new impossibility claim, find the freedom and the escape.