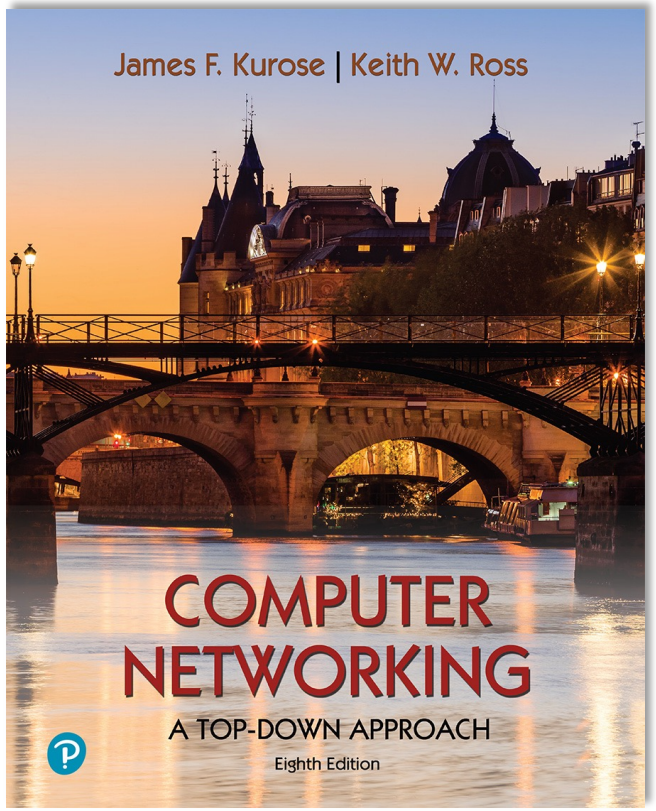
Chapter 6
The Link Layer and LANs

Yaxiong Xie

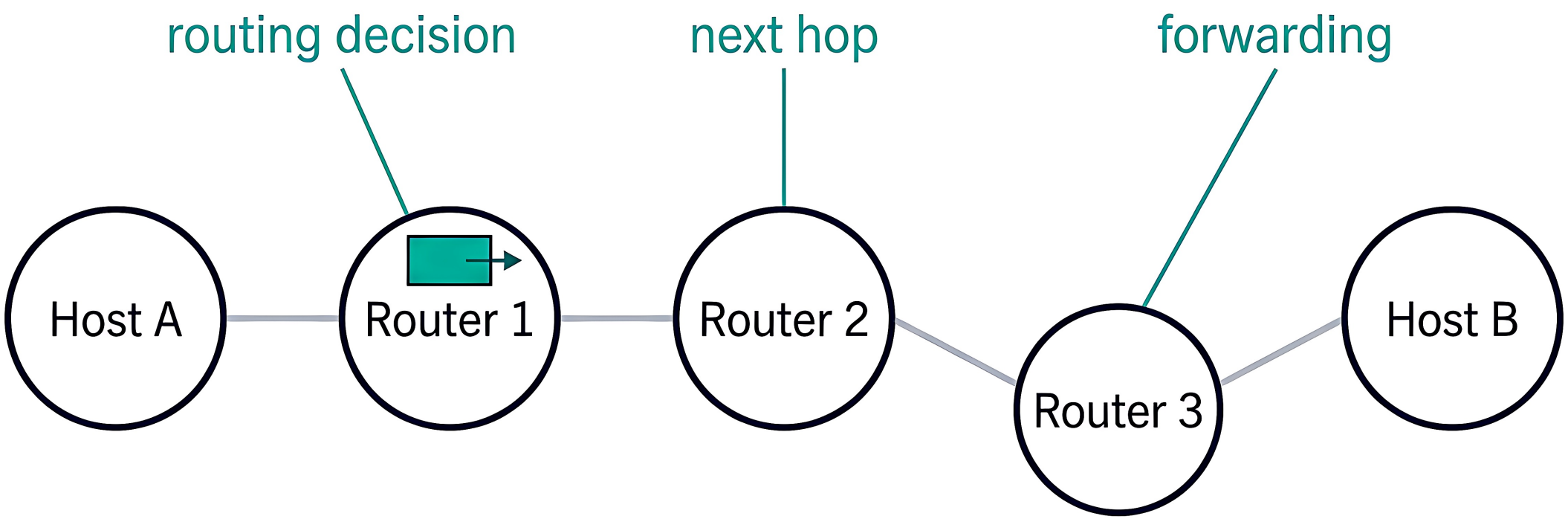
Department of Computer Science and Engineering
University at Buffalo, SUNY



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Adapted from the slides of the book's authors

The Global View: End-to-End Packet Delivery



At the network layer, our viewpoint is global. Routers examine the destination IP address and decide the next hop.

The shift in focus from the network layer to the link layer.

Global / Multi-hop

Scope: Entire network.

Goal: Reach the final destination.

Main Question: *Where should the packet go next?*

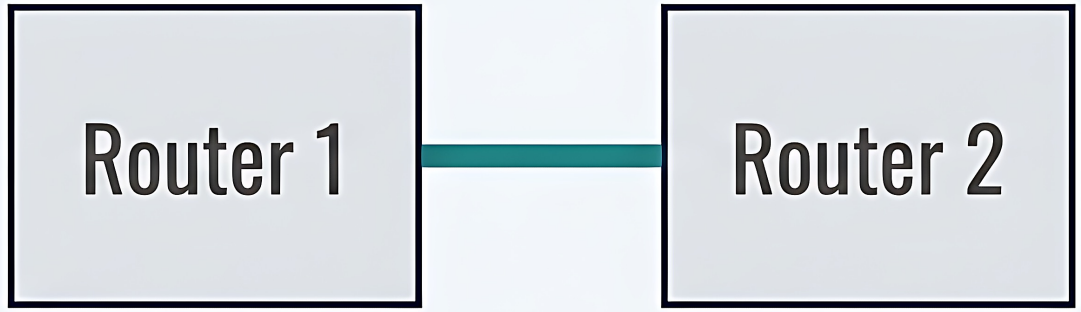
The shift in focus from the network layer to the link layer.

Global / Multi-hop

Scope: Entire network.

Goal: Reach the final destination.

Main Question: *Where should the packet go next?*



Local / One hop

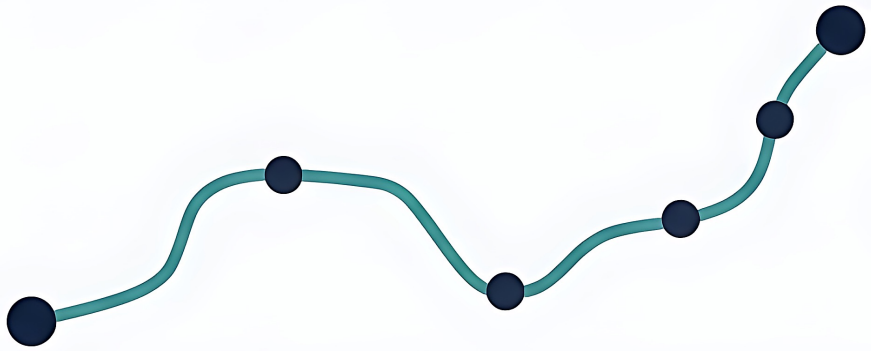
Scope: One specific physical link.

Goal: Deliver data to the immediate next node.

Main Question: *How do we send data over this physical medium?*

Trip Planning vs. Driving the Road Segment

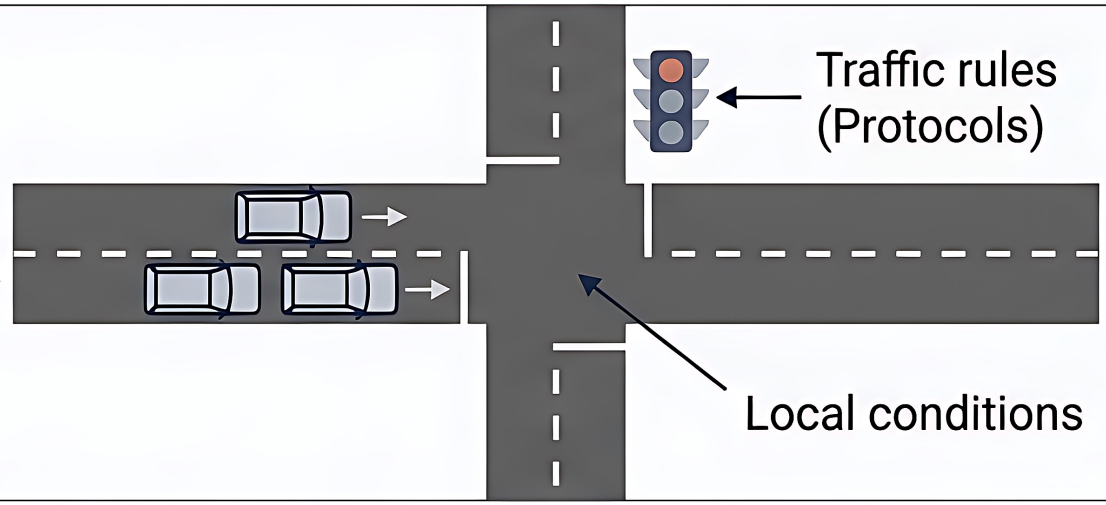
Network Layer Analogy



Plan overall route

Sequence of stops

Link Layer Analogy



Traffic rules
(Protocols)

Lane sharing
(Medium access)

Local conditions

The network layer plans the trip. The link layer handles the physical mechanics of driving a single road segment.

The Diversity of Physical Links

Point-to-Point Link

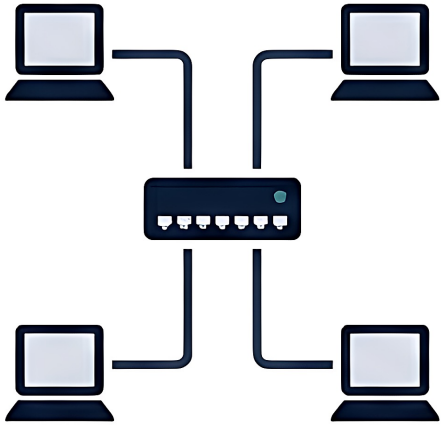
Dedicated

Not all links behave the same way. The link layer adapts abstract data to survive vastly different physical realities.

The Diversity of Physical Links

Point-to-Point Link

Dedicated



Wired Ethernet LAN

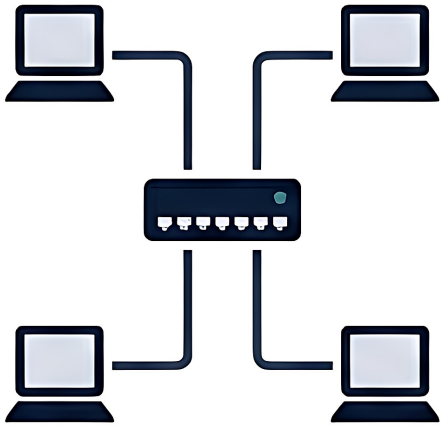
Shared

Not all links behave the same way. The link layer adapts abstract data to survive vastly different physical realities.

The Diversity of Physical Links

Point-to-Point Link

Dedicated



Wired Ethernet LAN

Shared

Wireless Wi-Fi

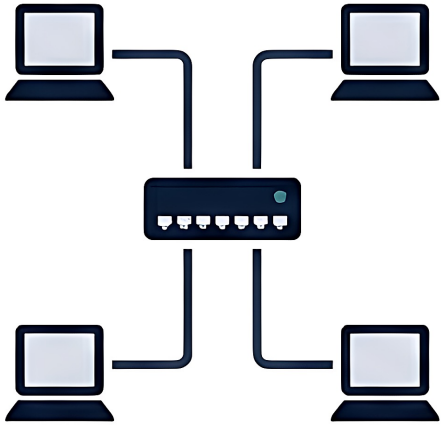
Contention & Noisy

Not all links behave the same way. The link layer adapts abstract data to survive vastly different physical realities.

The Diversity of Physical Links

Point-to-Point Link

Dedicated

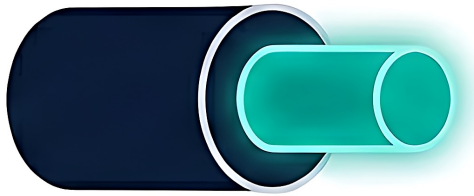


Wired Ethernet LAN

Shared

Wireless Wi-Fi

Contention & Noisy

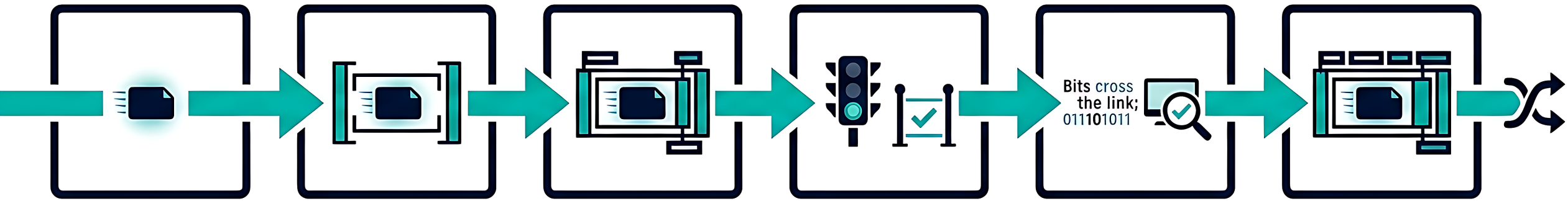


Long-haul Fiber

Physical distance

Not all links behave the same way. The link layer adapts abstract data to survive vastly different physical realities.

The Link Layer Process Pipeline



Input
Network-layer packet arrives.

Encapsulation
Packet is packaged into a structural frame.

Addressing
Local link receiver identity is attached.

Medium Access
Control logic determines when it is safe to transmit.

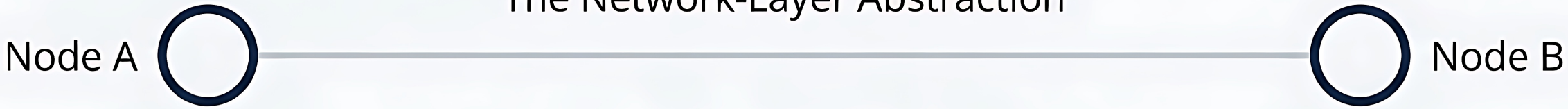
Transmission & Check
Bits cross the link; receiver detects or corrects bit errors.

Output
Data delivered to the next node.

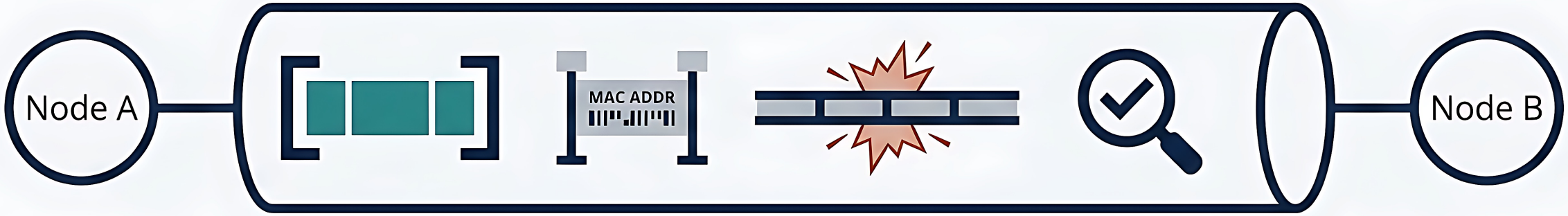
The link layer turns the abstract idea of a ‘next hop’ into an actual physical transmission sequence.

The black box we are opening in this chapter.

The Network-Layer Abstraction



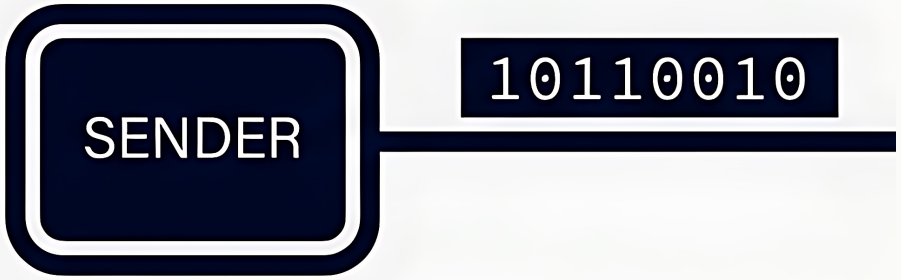
Previously, we treated links as simple edges in a graph.
Now, we open up that black box.



The Link-Layer Reality

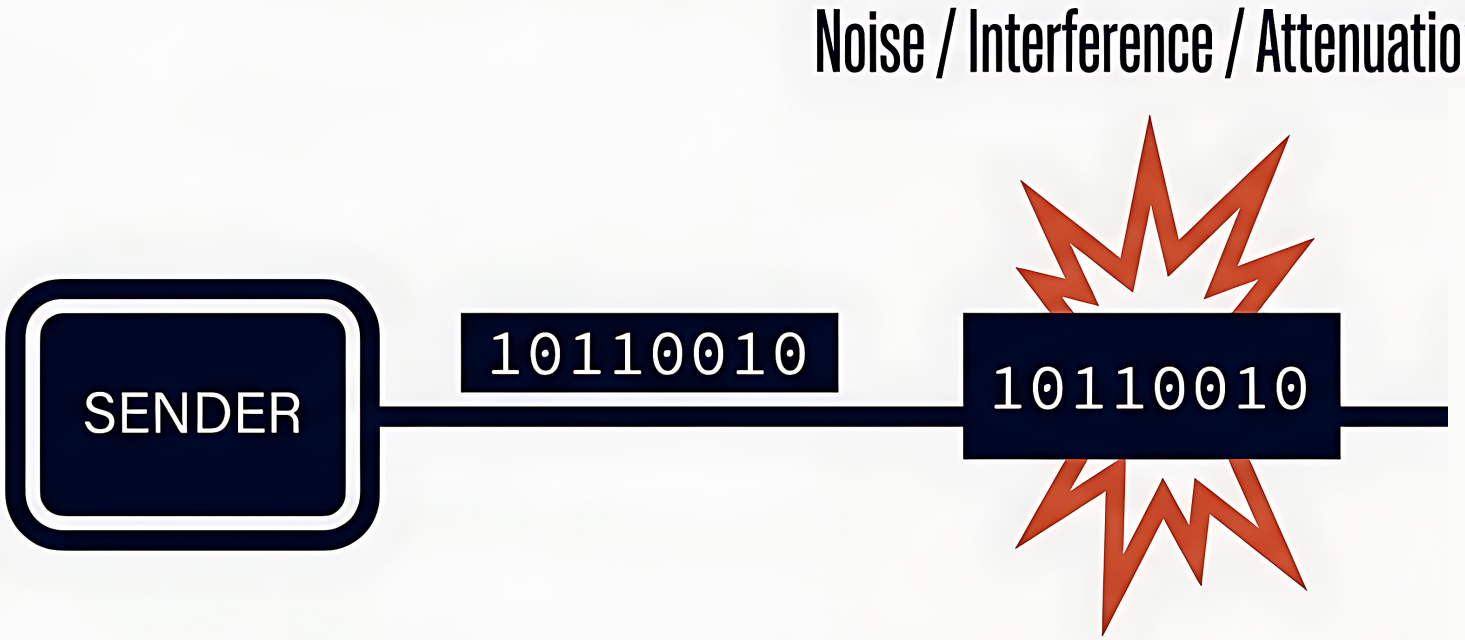
Error Detection

TRANSMISSION ERRORS ARE A PHYSICAL REALITY



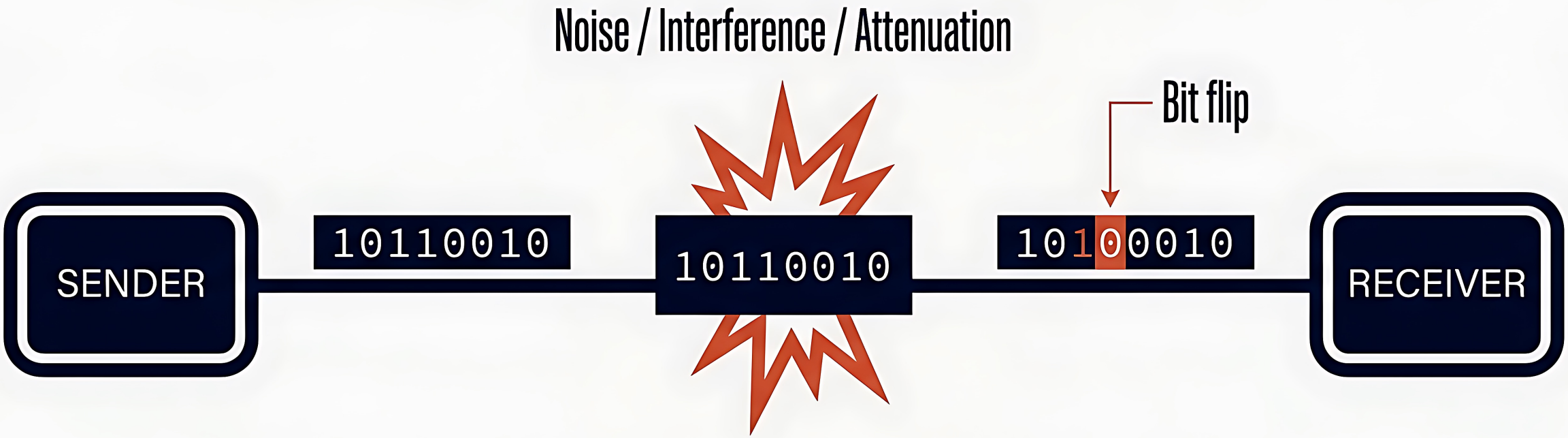
Communication links are not perfect. The question is not whether errors happen, but how the system notices them.

TRANSMISSION ERRORS ARE A PHYSICAL REALITY



Communication links are not perfect. The question is not whether errors happen, but how the system notices them.

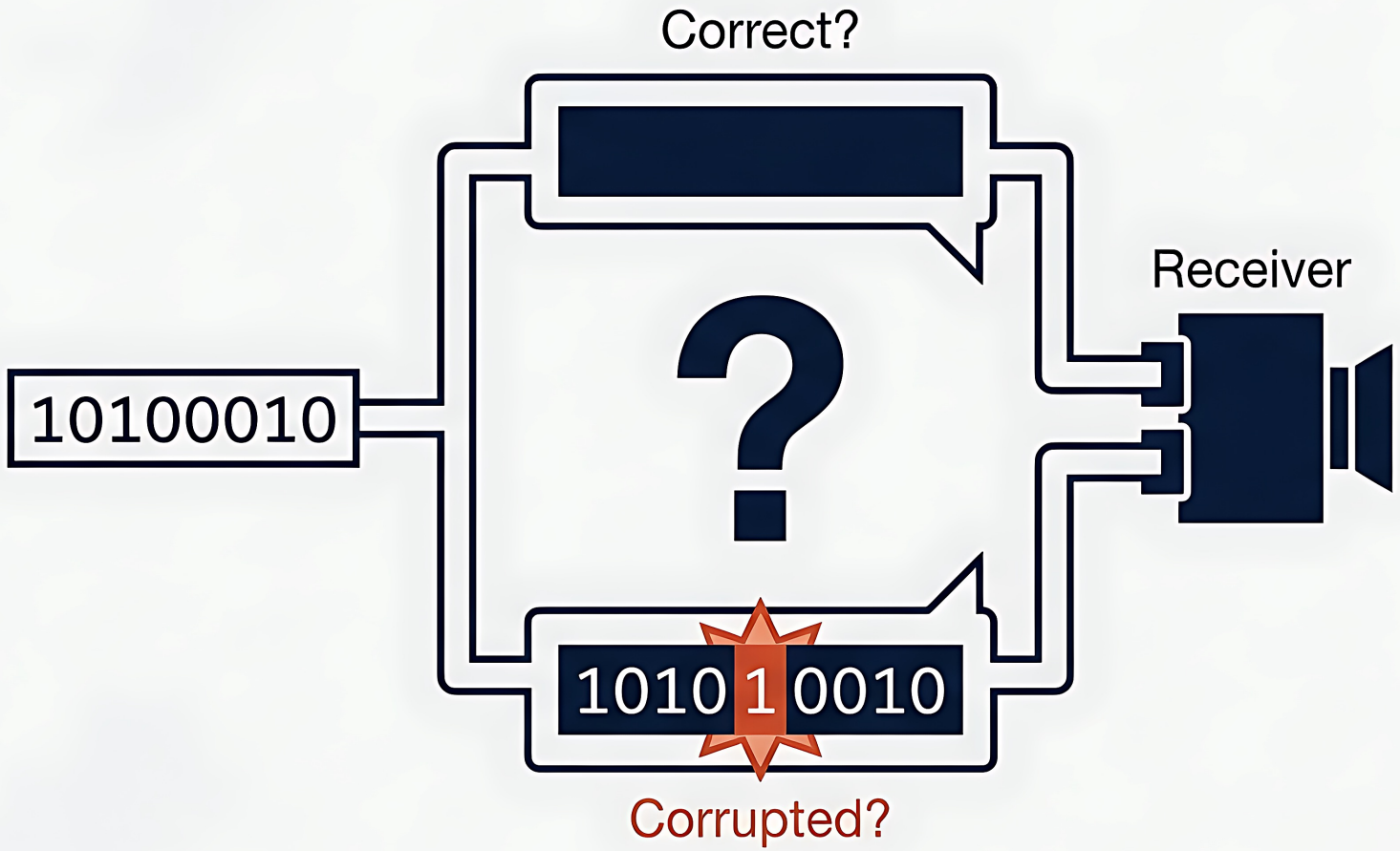
TRANSMISSION ERRORS ARE A PHYSICAL REALITY



Communication links are not perfect. The question is not whether errors happen, but how the system notices them.

Raw data alone cannot prove its own integrity

- If the receiver sees only the raw bits, it has no reference point.
- The sequence might be the original data, or a corrupted version.
- The receiver cannot tell the difference.

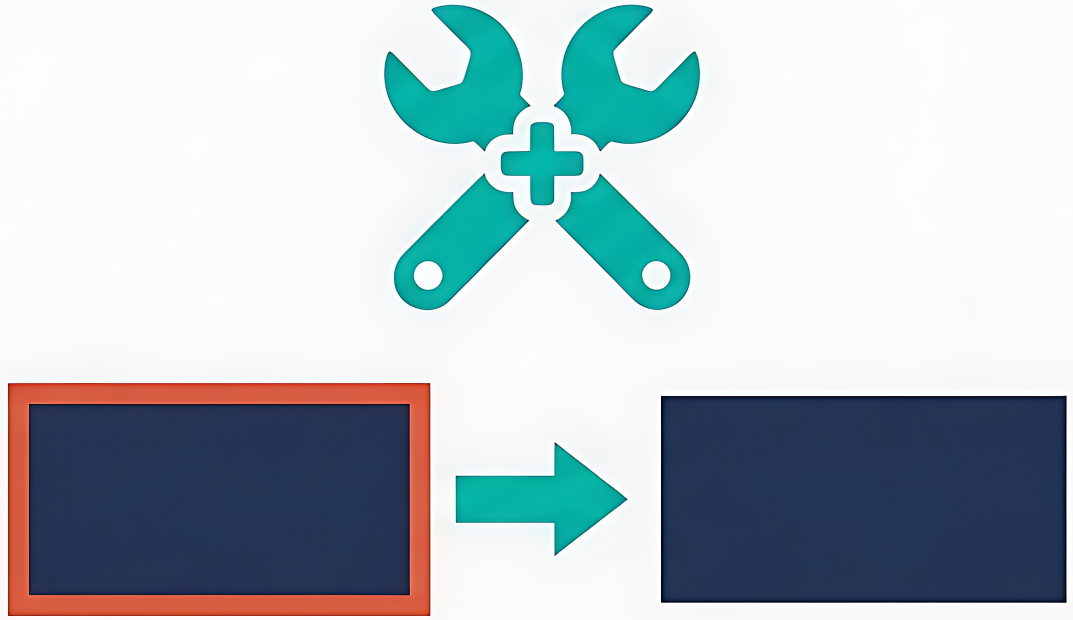


Knowing a frame is broken is different from fixing it

Error Detection

The receiver knows it is wrong.

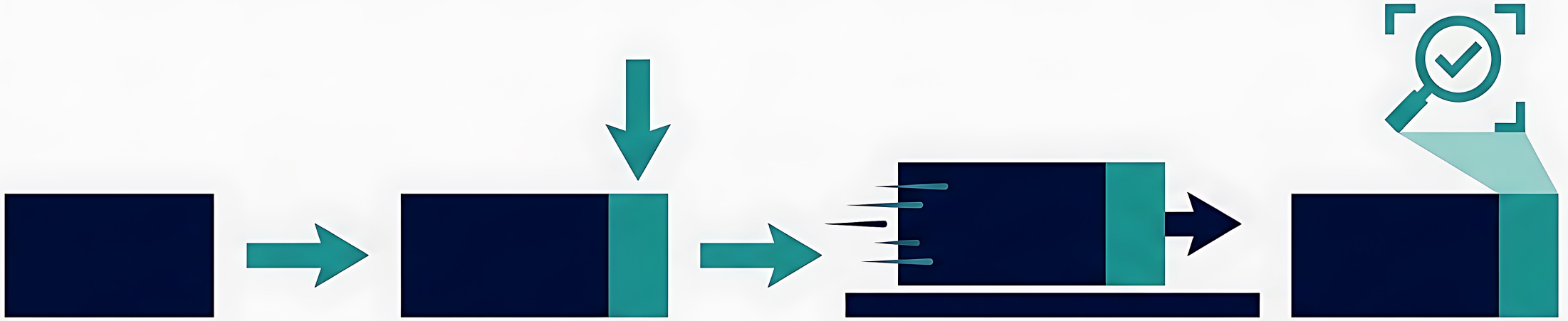
Error Correction



The receiver recovers the original data.

In this lecture, our focus is strictly on detection.

Senders must attach extra structured information



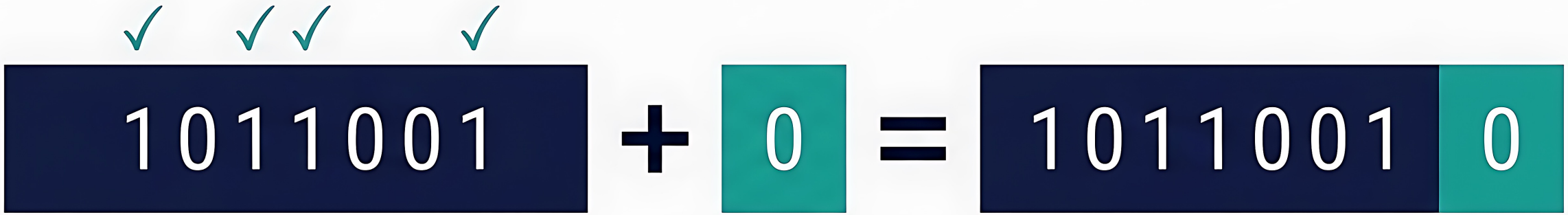
Original Data

Add extra check bits
(Redundancy)

Receiver verifies

The common principle behind all error-detection schemes is adding redundancy so the receiver can check validity.

Parity uses a single extra bit to enforce an even count



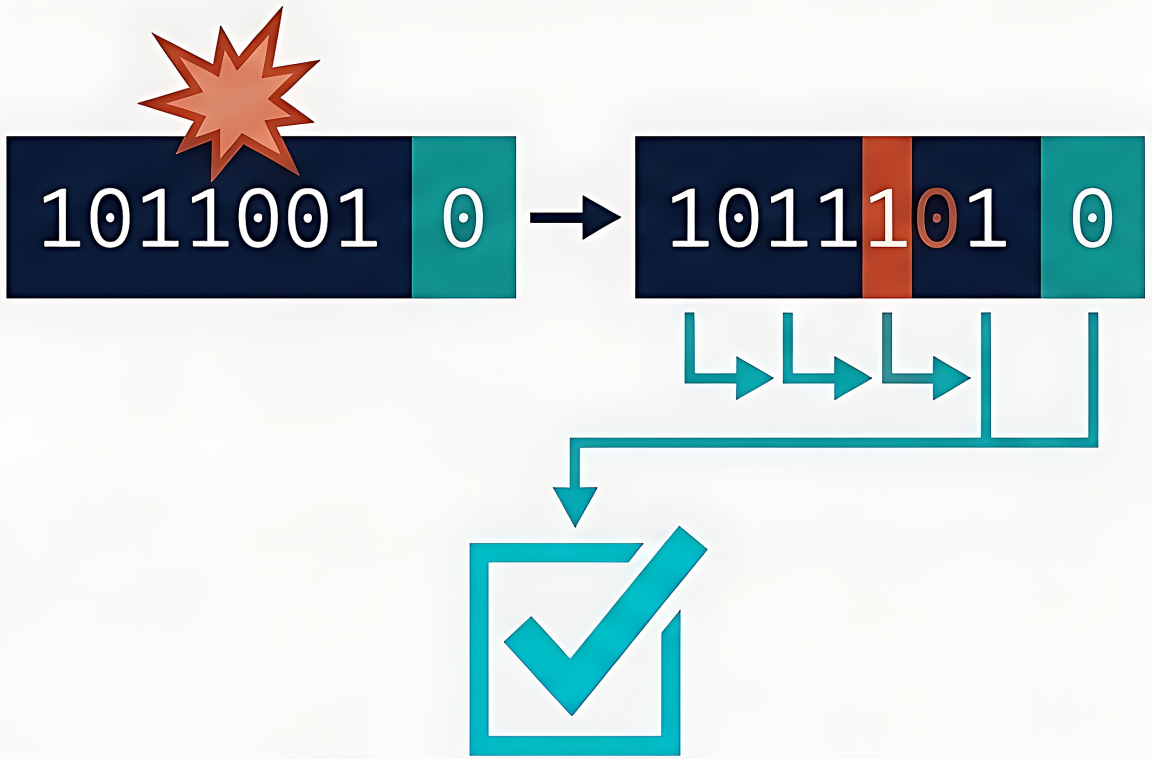
Data: Count the 1s (Total: 4)

Add parity bit

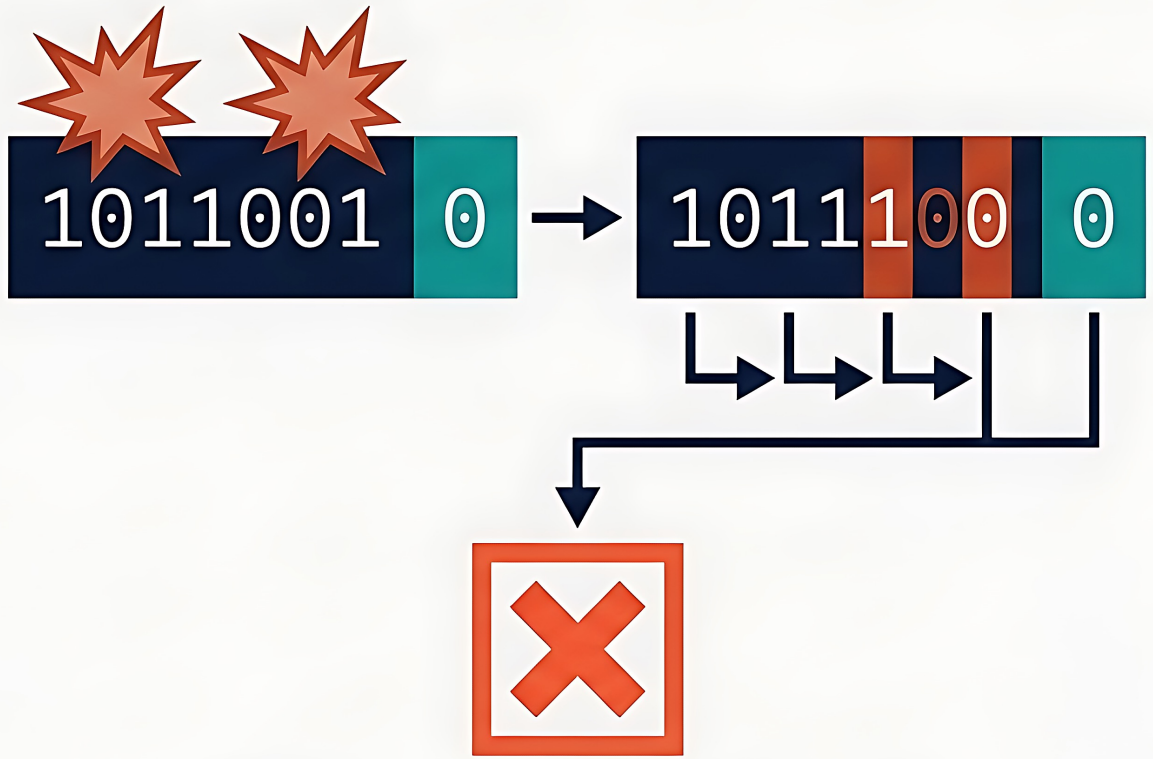
Even Parity Rule

The sender adds one extra bit so that the total number of 1 1s follows a strict mathematical rule.

A simple parity check completely fails if two bits flip



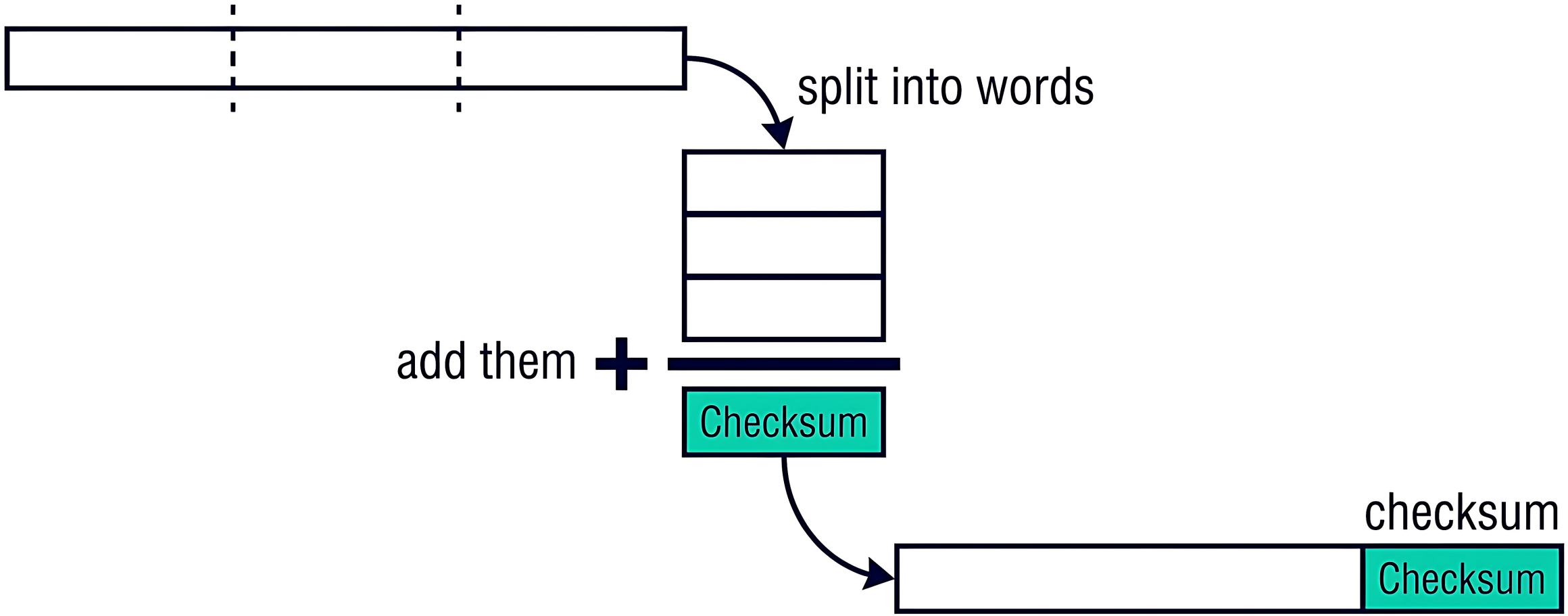
1-bit error → Detected



2-bit error → Slips through

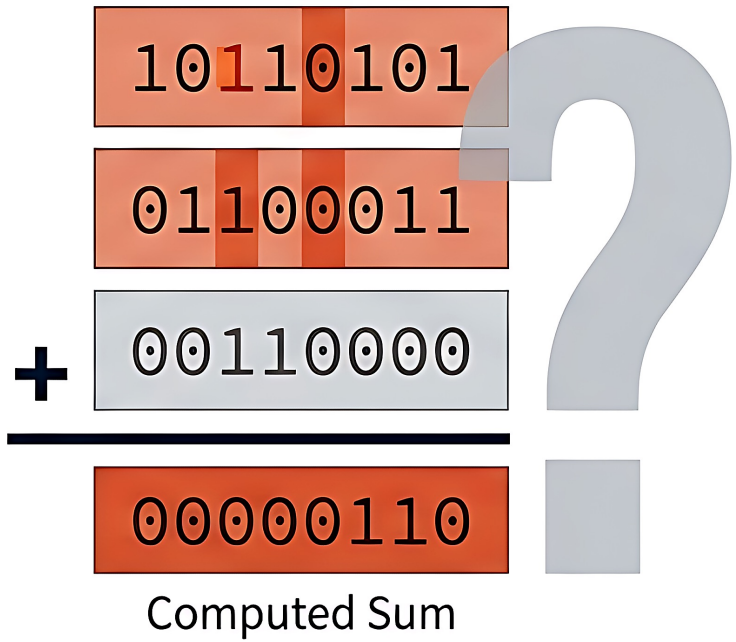
Parity is cheap, but weak. It cannot detect all patterns of corruption.

Checksum uses arithmetic summarization instead of a single parity rule



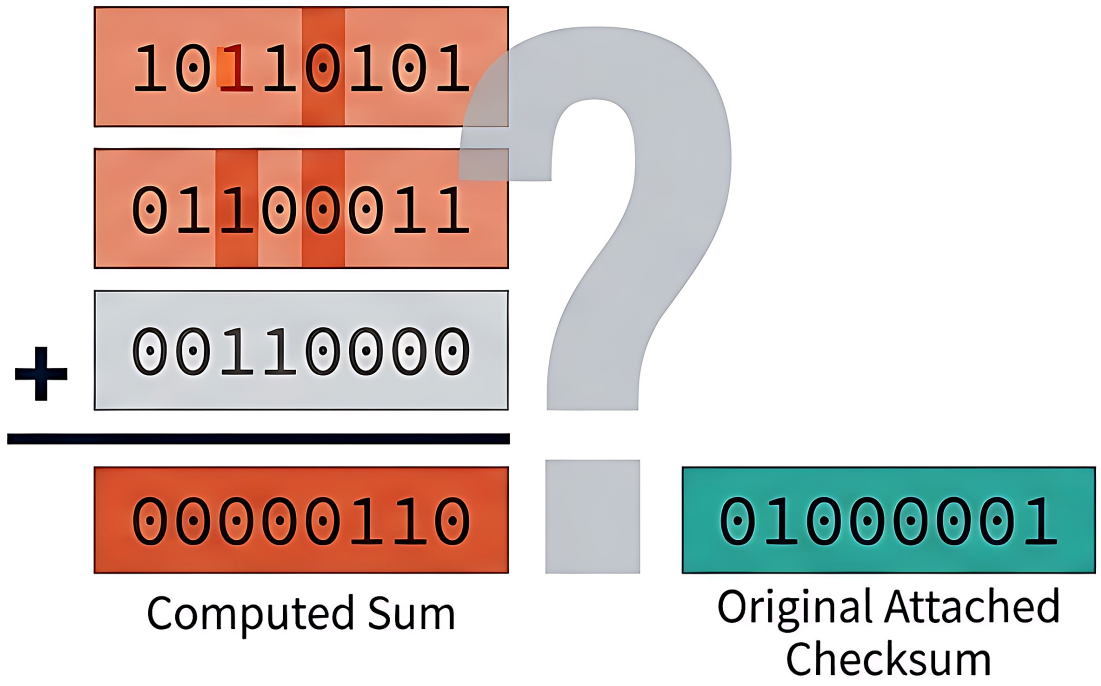
Checksum is still detection, not correction

A mismatch signals corruption, but what was the correct data? Checksums offer no path to recovery.



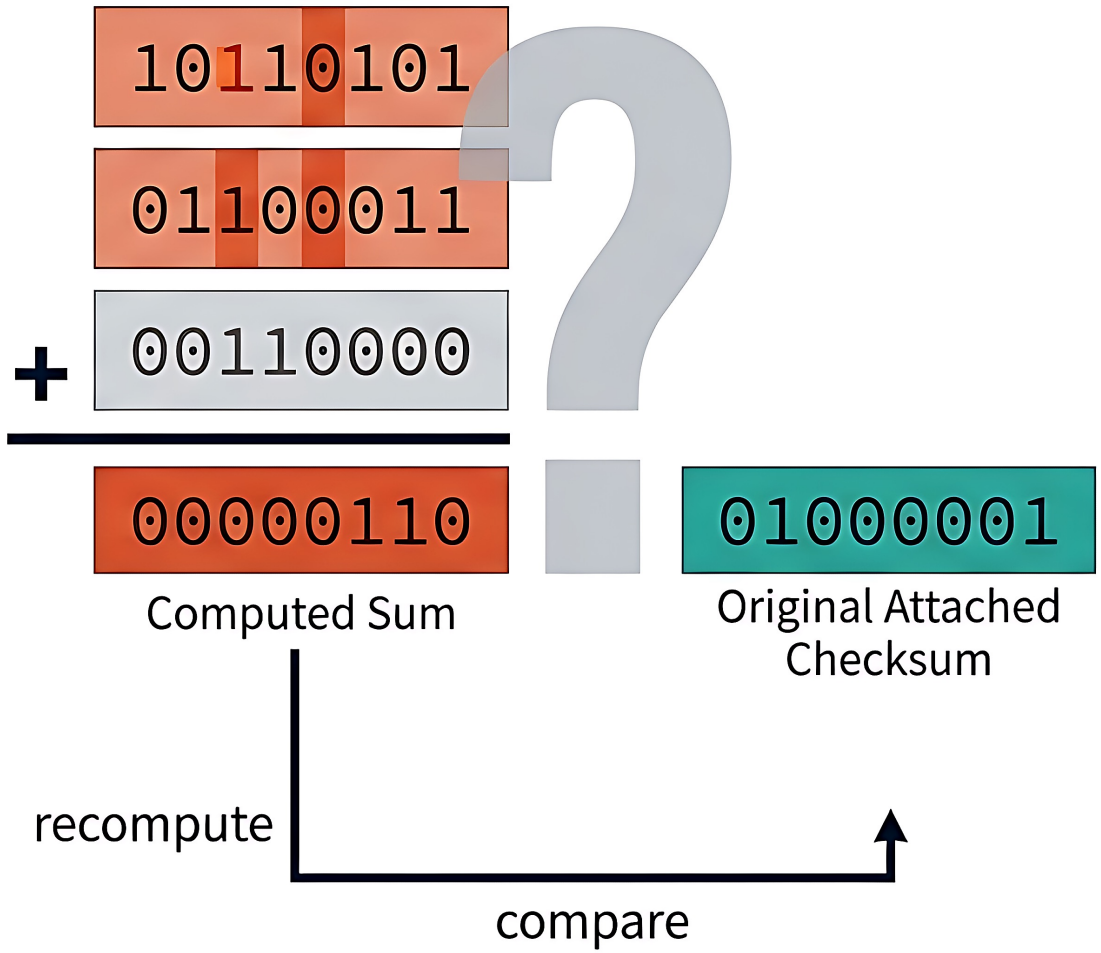
Checksum is still detection, not correction

A mismatch signals corruption, but what was the correct data? Checksums offer no path to recovery.



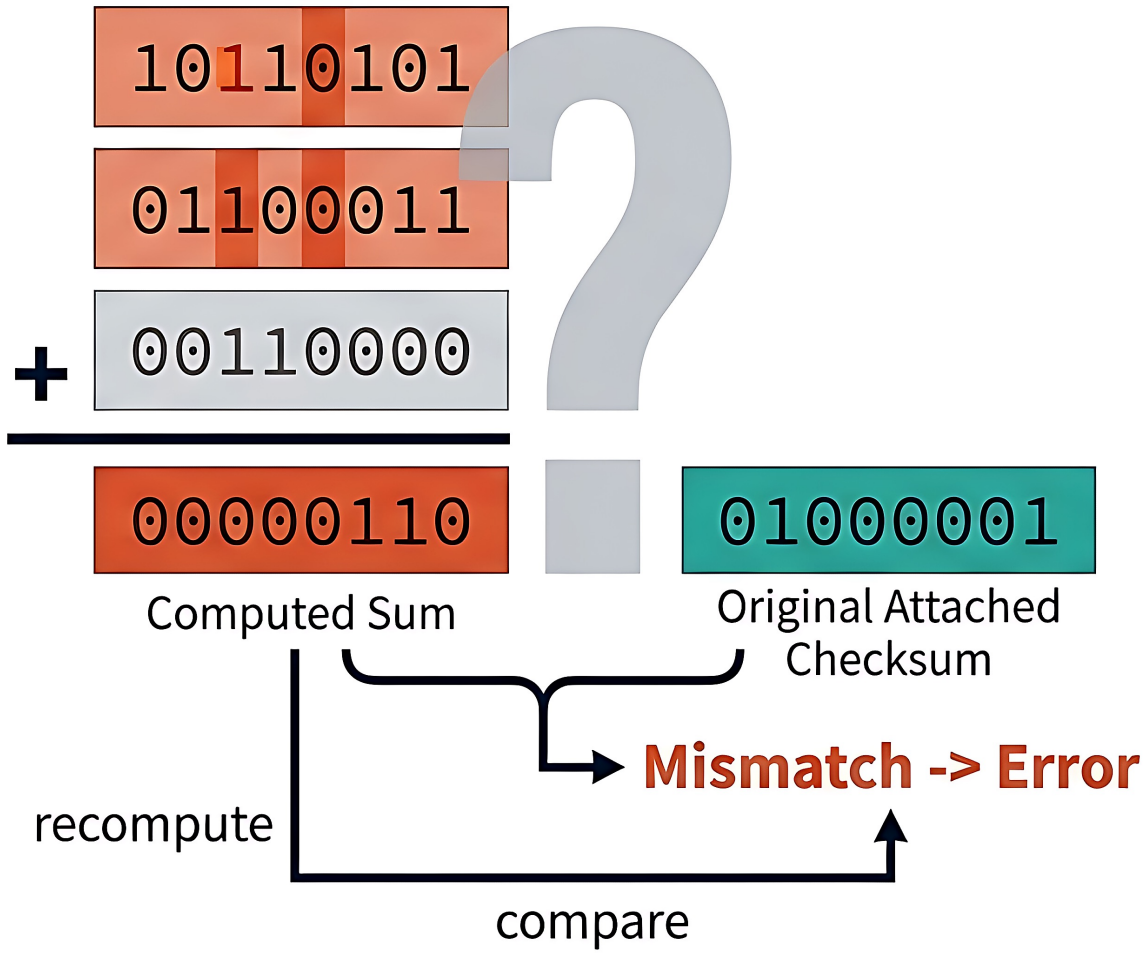
Checksum is still detection, not correction

A mismatch signals corruption, but what was the correct data? Checksums offer no path to recovery.



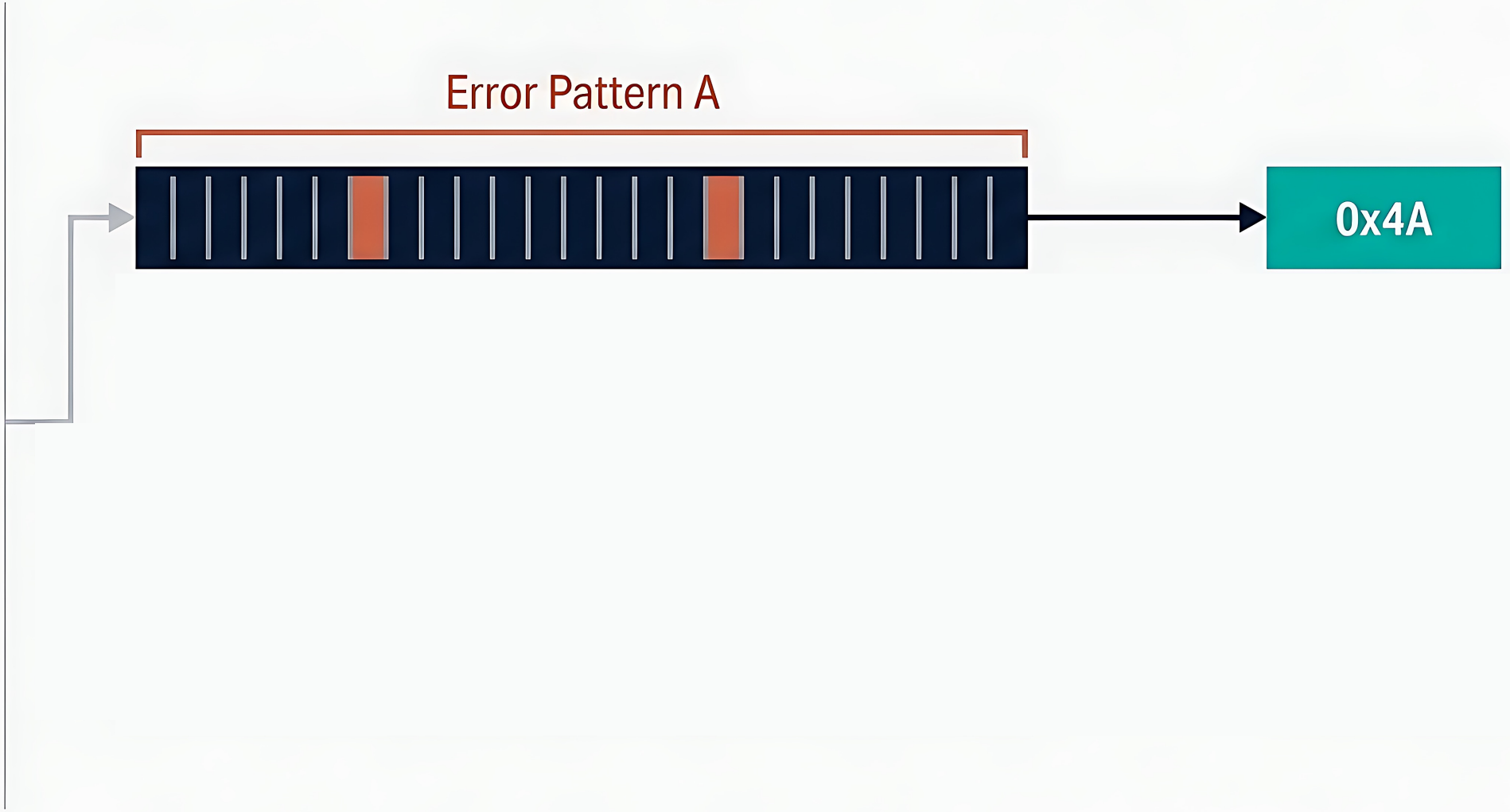
Checksum is still detection, not correction

A mismatch signals corruption, but what was the correct data? Checksums offer no path to recovery.



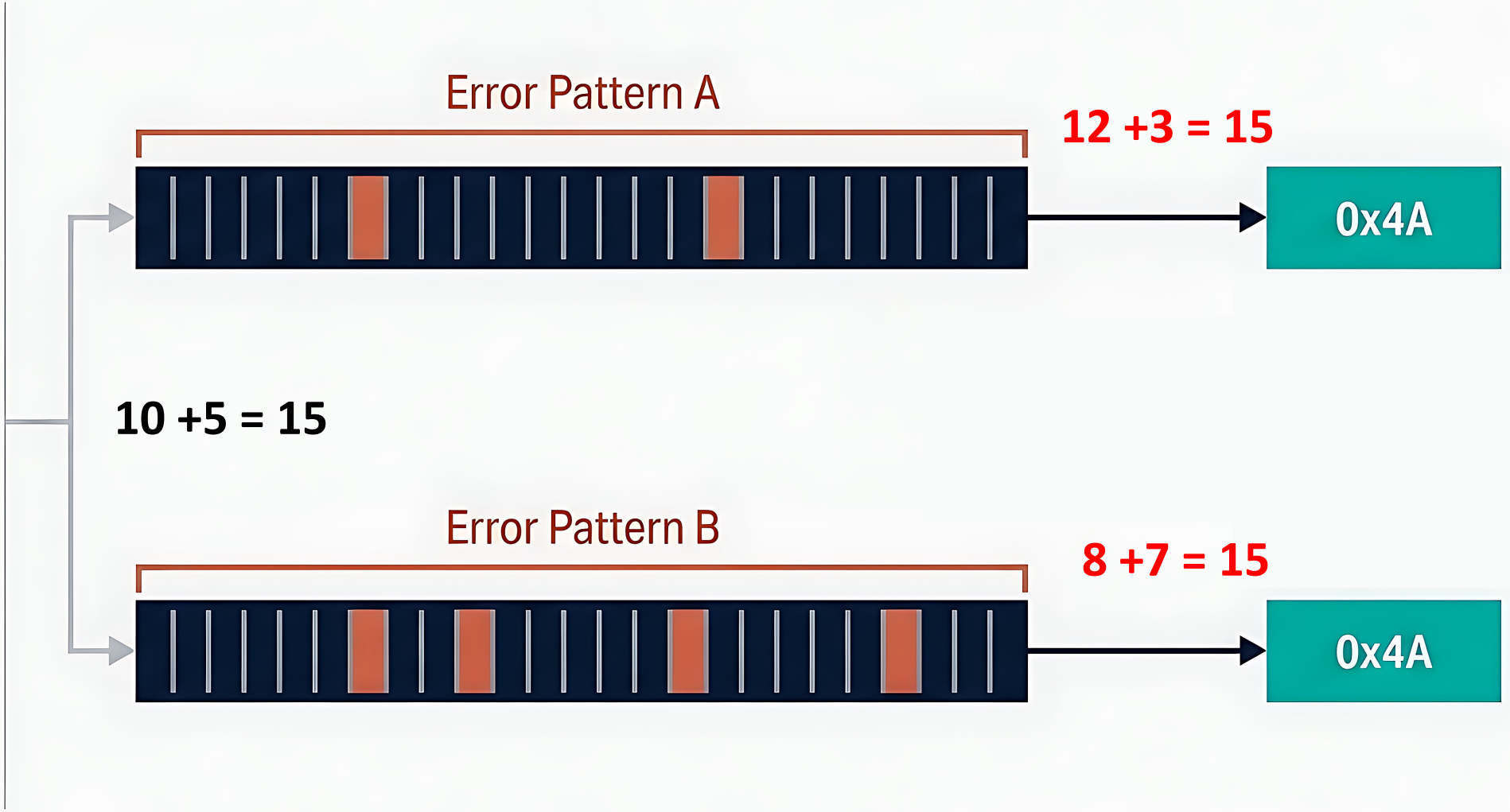
Different error patterns can produce the exact same checksum

Checksum is a practical improvement, but it is not perfect. It can miss specific combinations of errors.



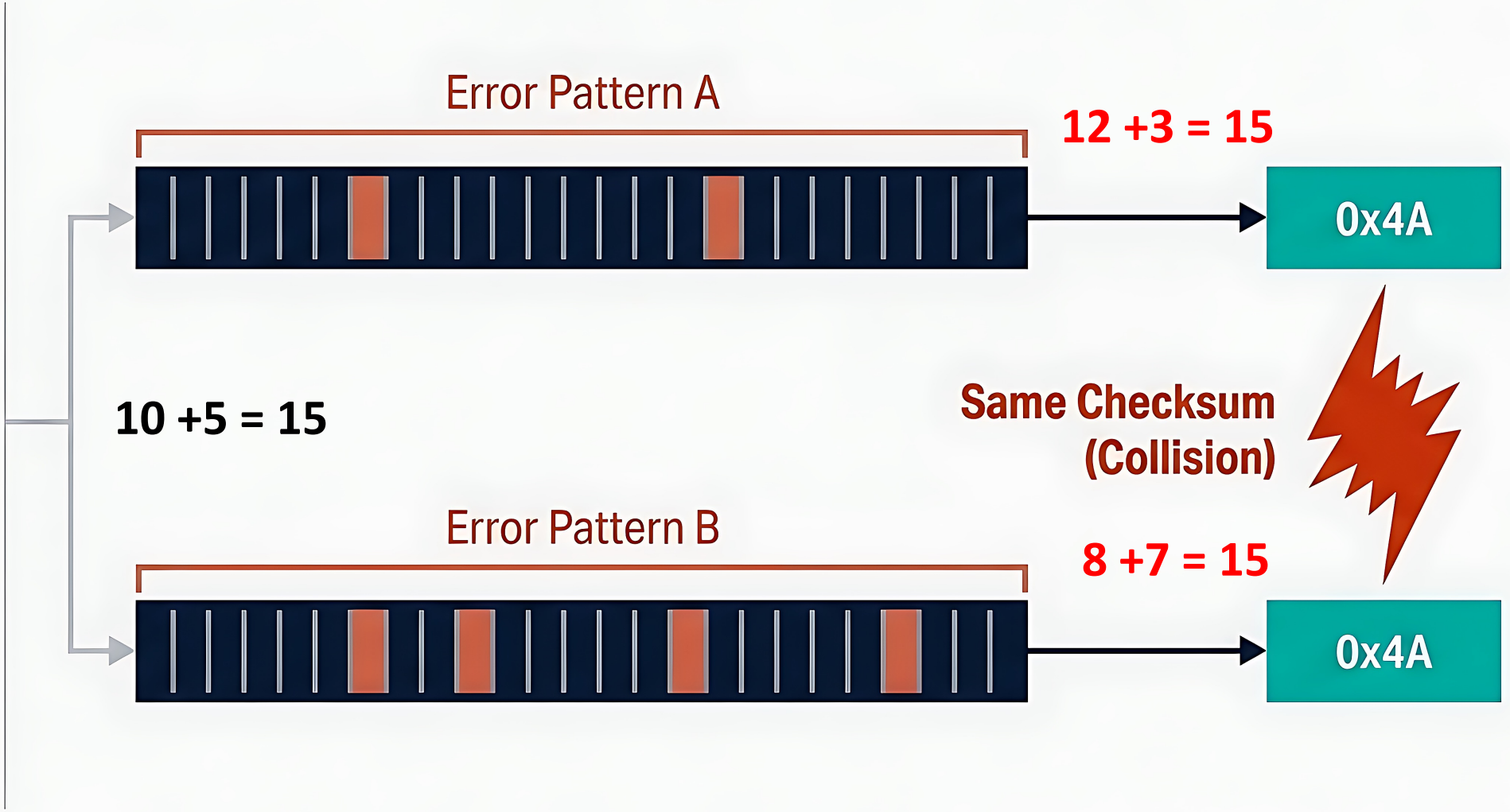
Different error patterns can produce the exact same checksum

Checksum is a practical improvement, but it is not perfect. It can miss specific combinations of errors.

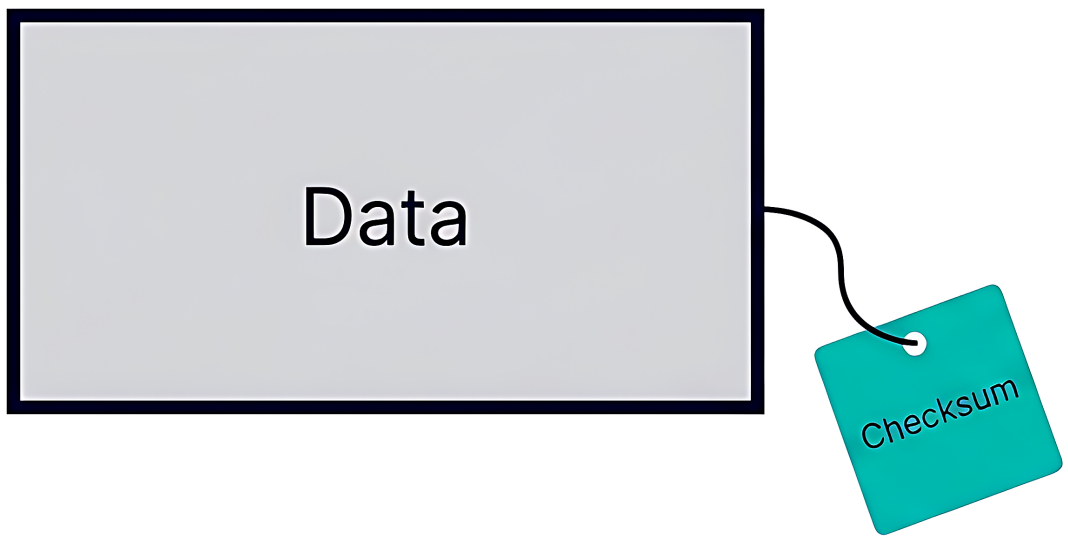


Different error patterns can produce the exact same checksum

Checksum is a practical improvement, but it is not perfect. It can miss specific combinations of errors.



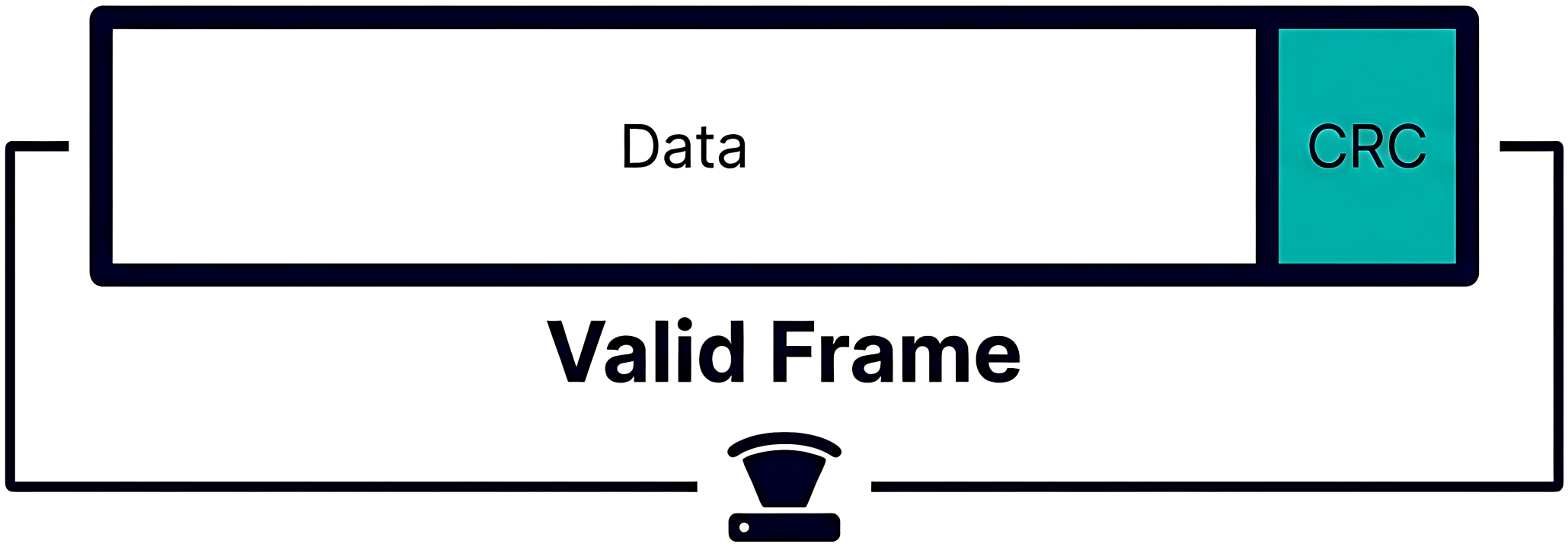
CRC is not just another checksum: it imposes a stronger structure on the transmitted bits



Checksum: Summarize.

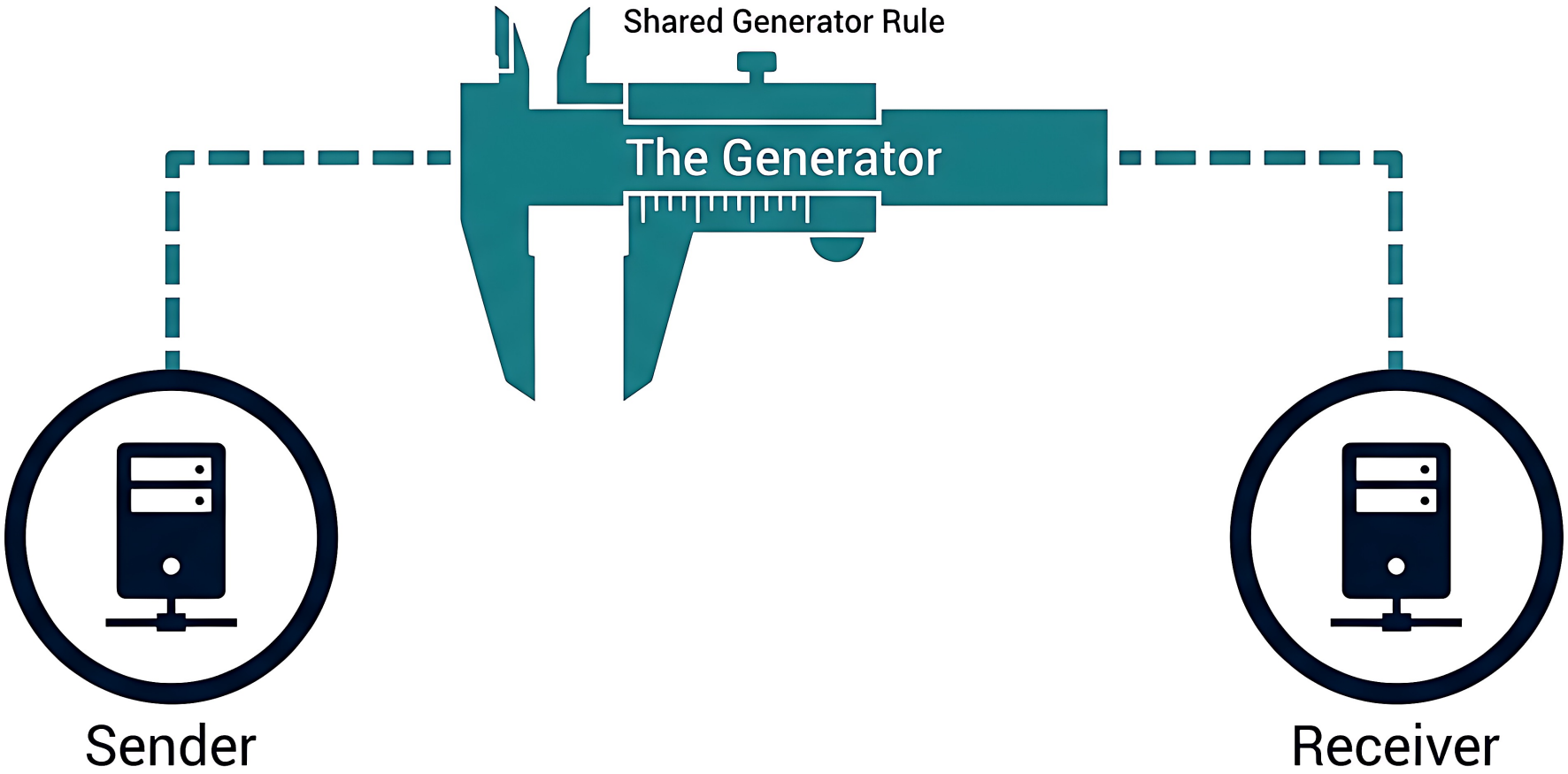
CRC: Enforce structure.
The whole frame is checked.

To understand CRC, we need to shift from “summary bits” to “valid codeword structure”



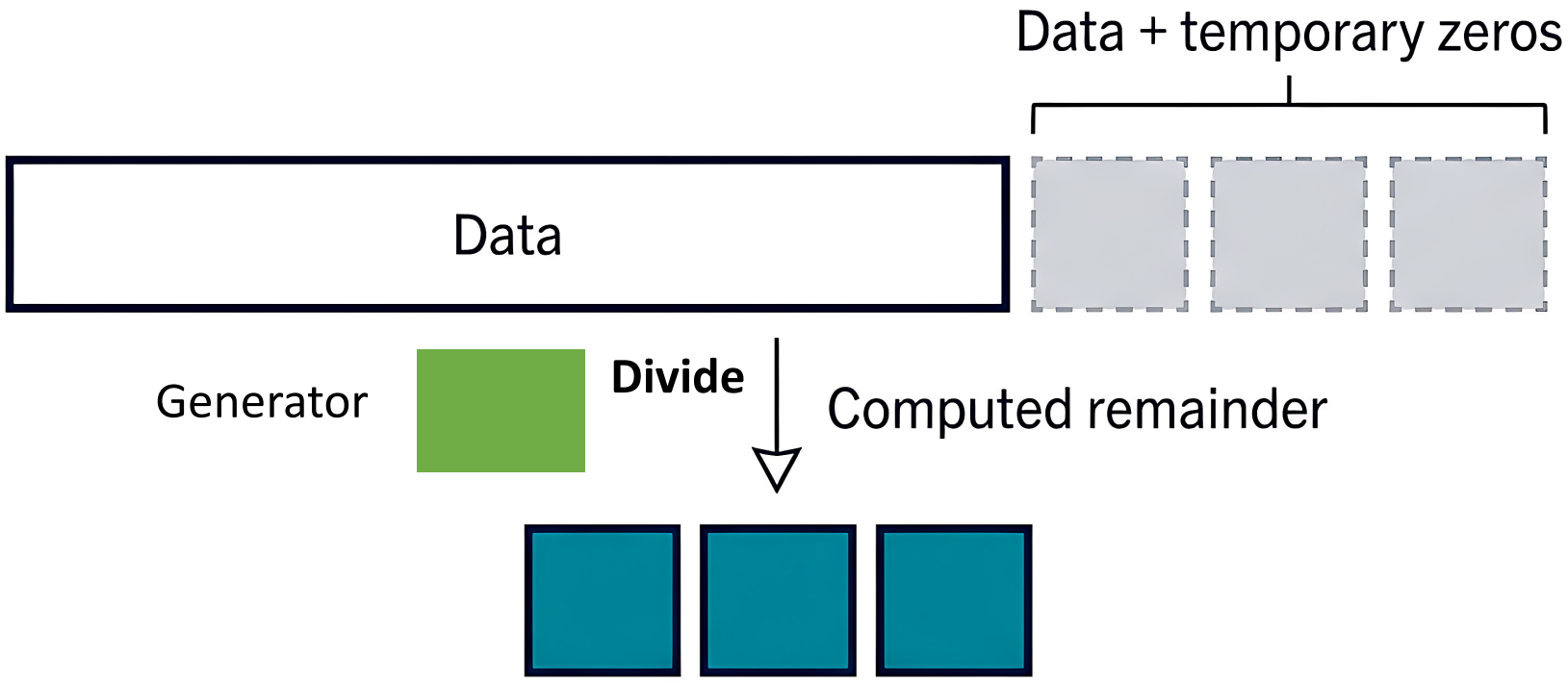
The receiver checks the whole thing.

The Foundation of CRC is a Pre-Shared Mathematical Rule

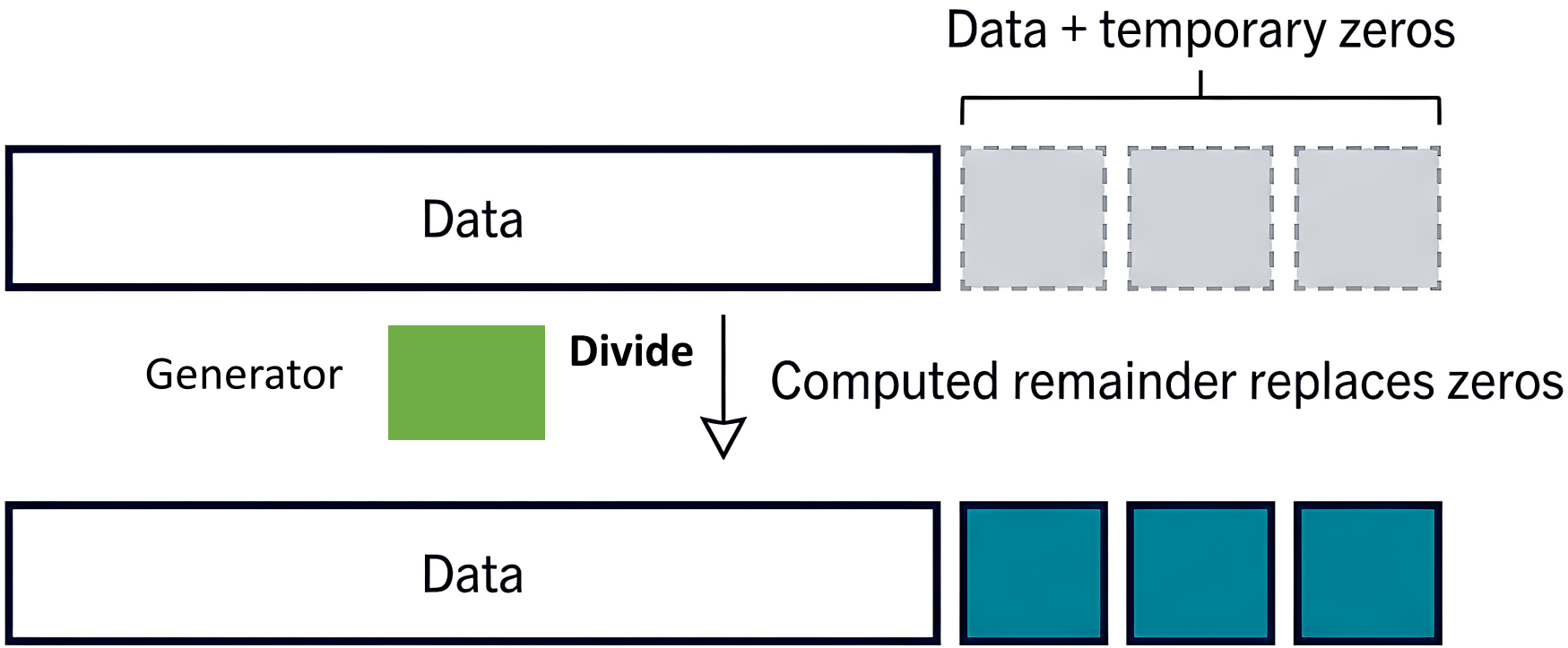


CRC works because both ends of the transmission agree on the exact same checking rule before any data is sent.

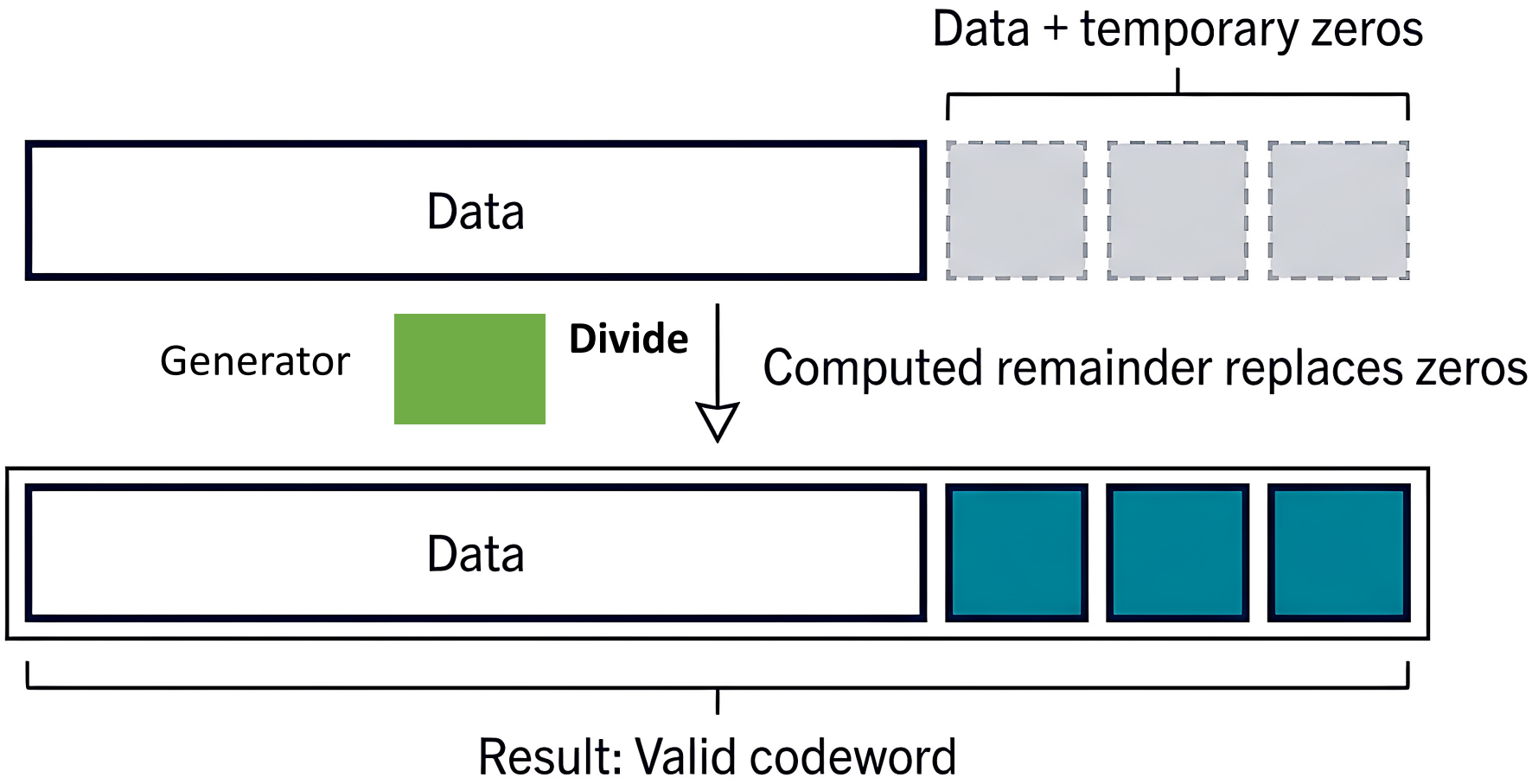
CRC Bits Are Carefully Computed



CRC Bits Are Carefully Computed

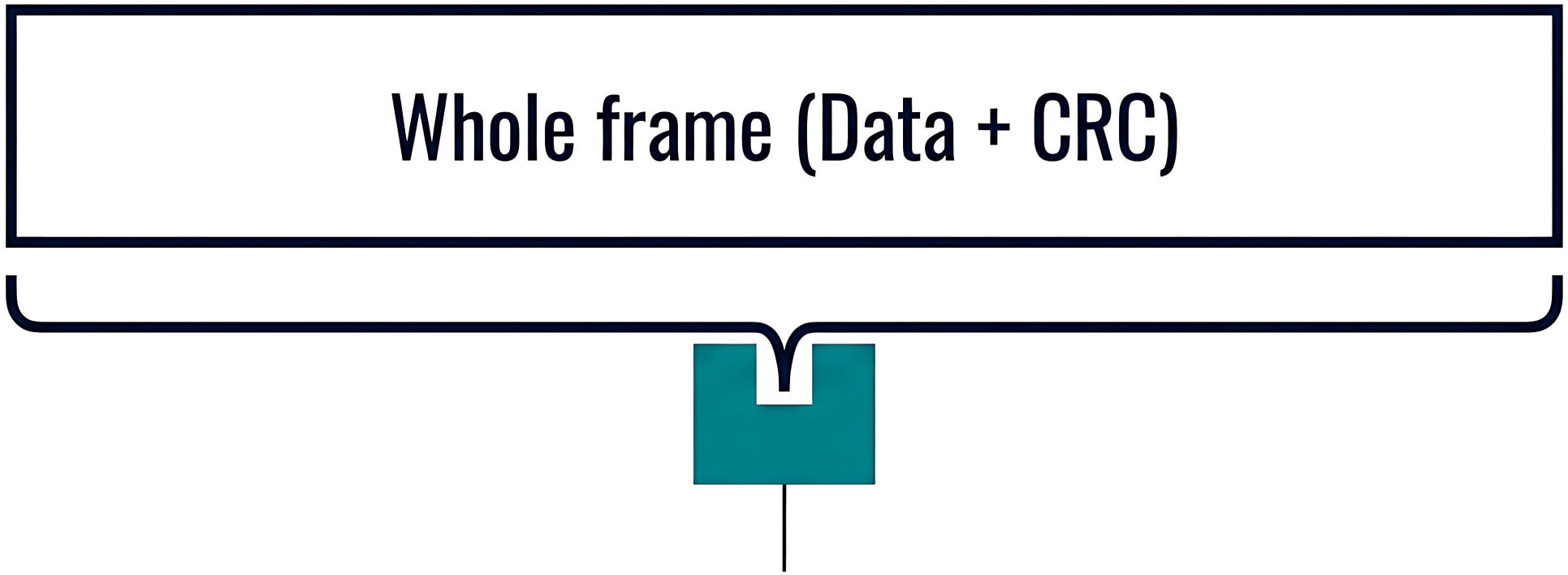


CRC Bits Are Carefully Computed



The Structural Core of CRC

The entire bit string is constructed to satisfy a strict divisibility condition.

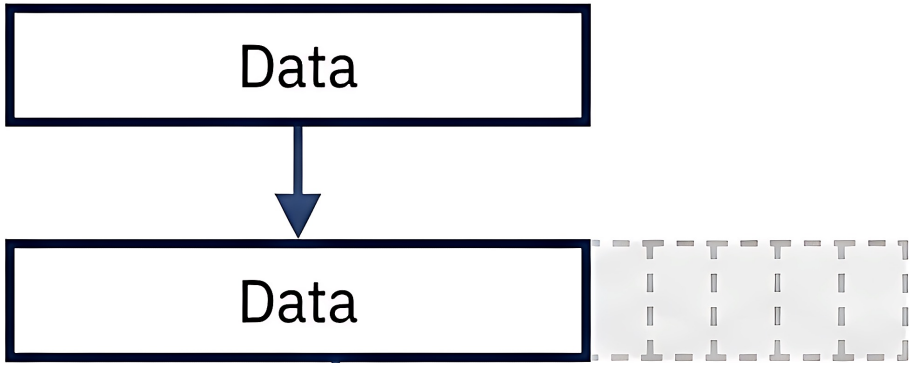


The Sender-Side Workflow

1. Start with data

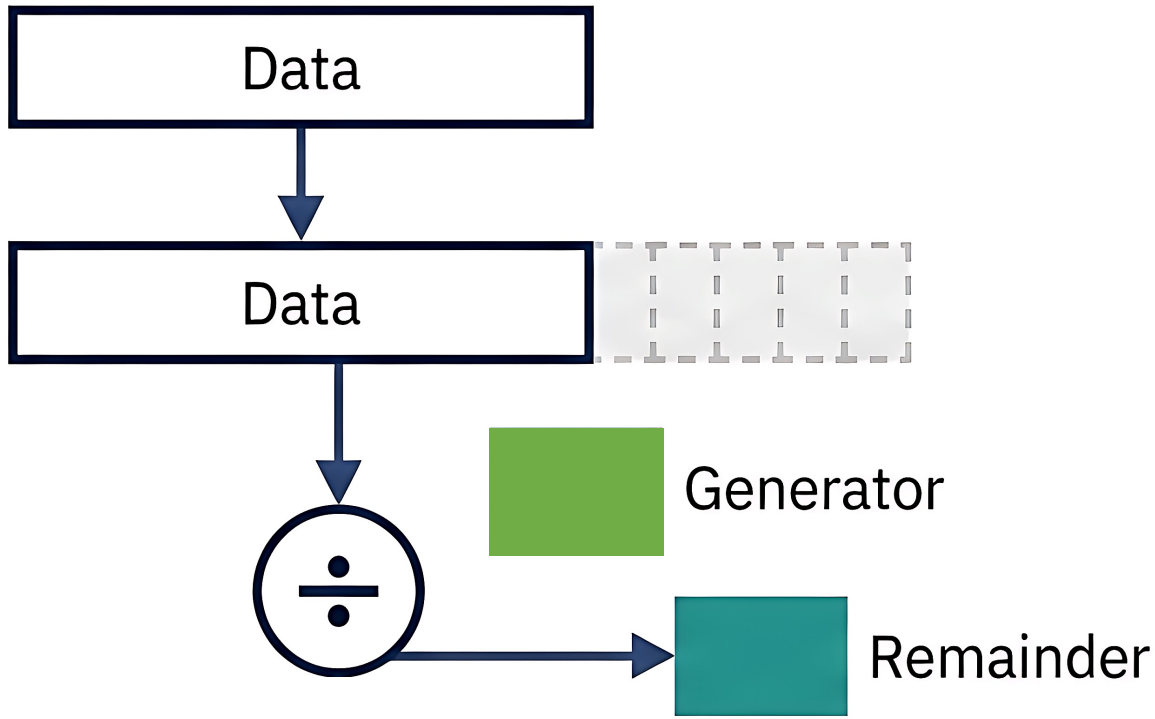
The Sender-Side Workflow

1. Start with data
2. Append placeholder zeros



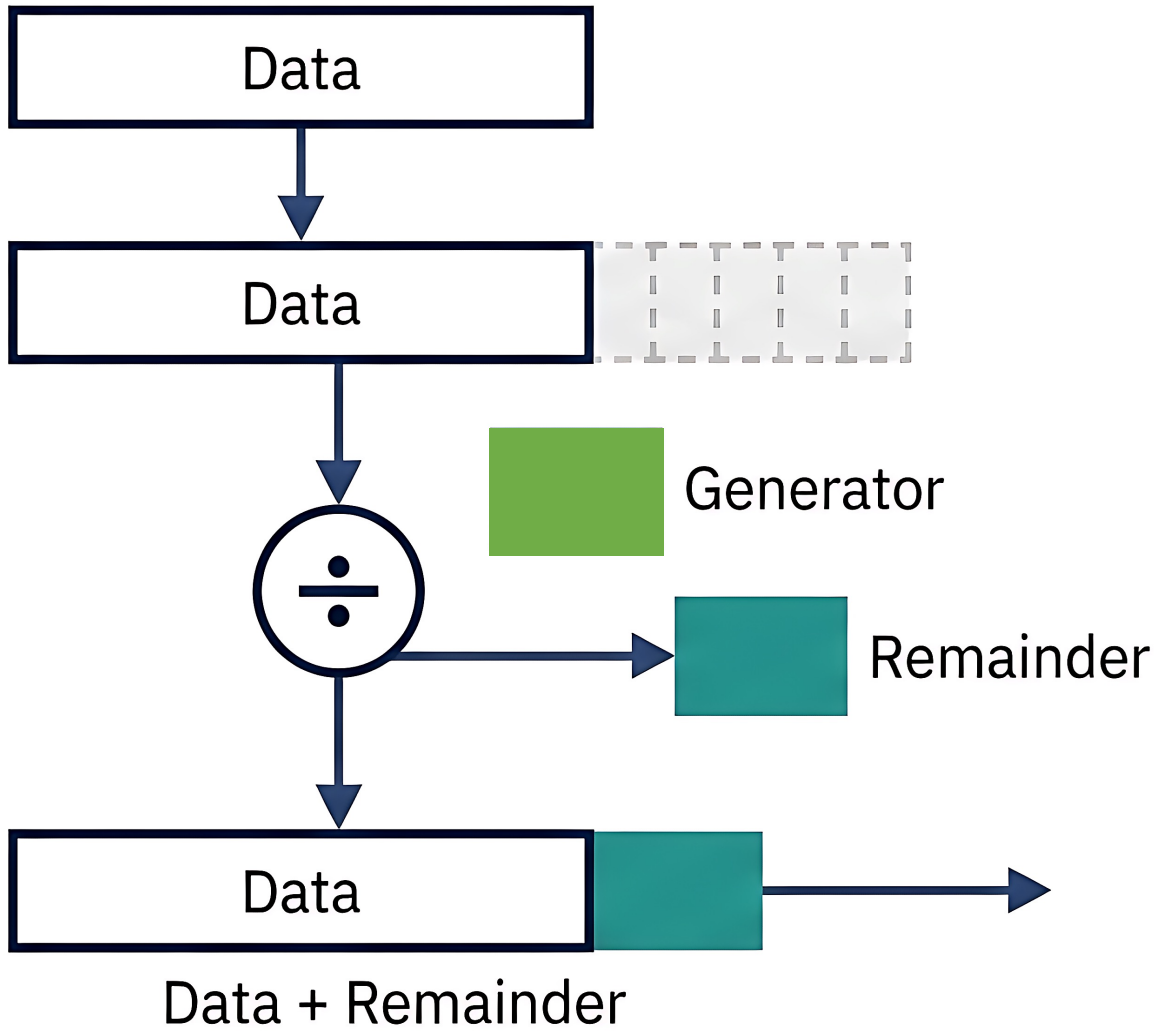
The Sender-Side Workflow

1. Start with data
2. Append placeholder zeros
3. Divide by generator to compute remainder



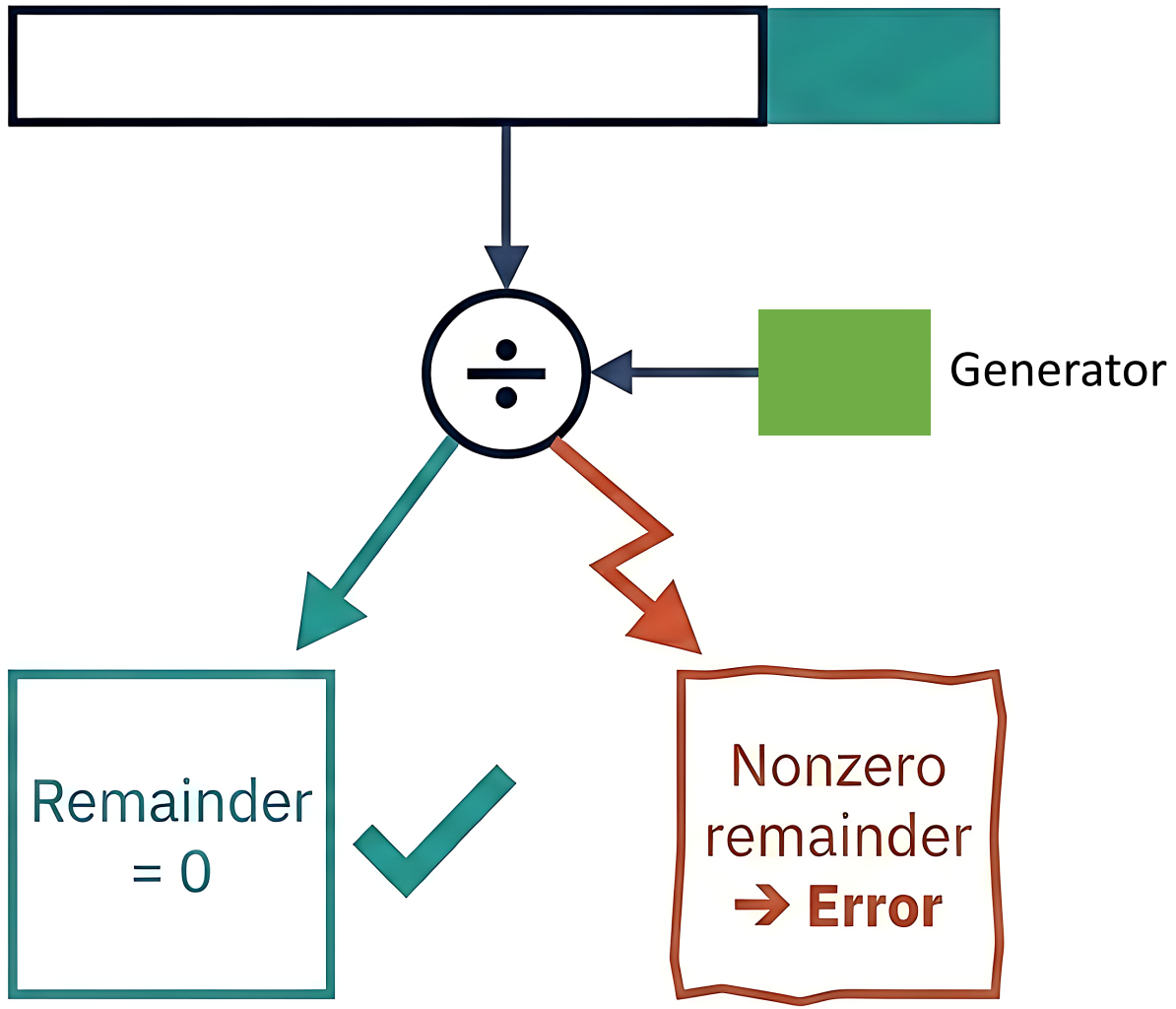
The Sender-Side Workflow

1. Start with data
2. Append placeholder zeros
3. Divide by generator to compute remainder
4. Transmit data + remainder

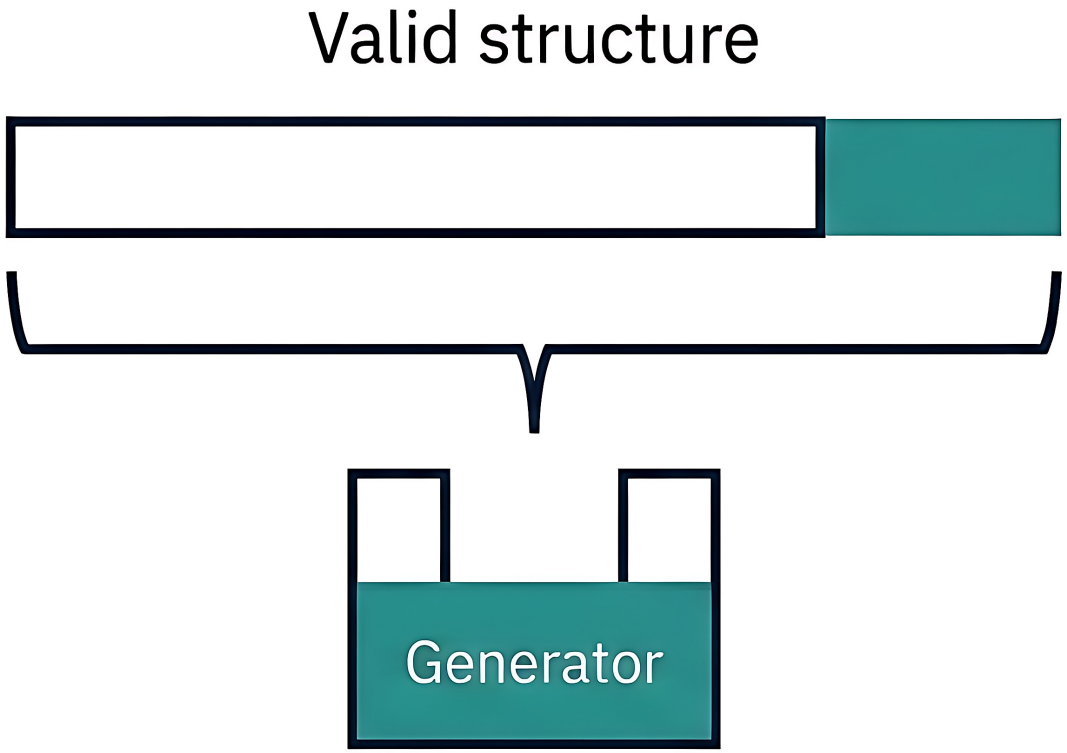


The Receiver-Side Verification

- Receive full frame
- Divide by the same generator
- Check the remainder



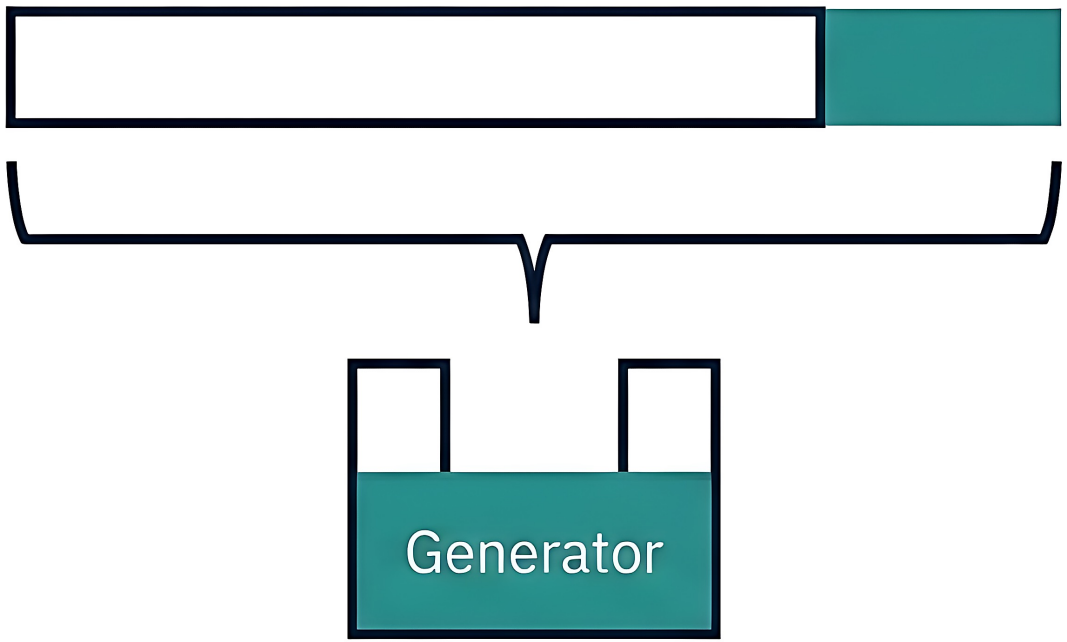
Errors Break the Structural Rule



Passes CRC

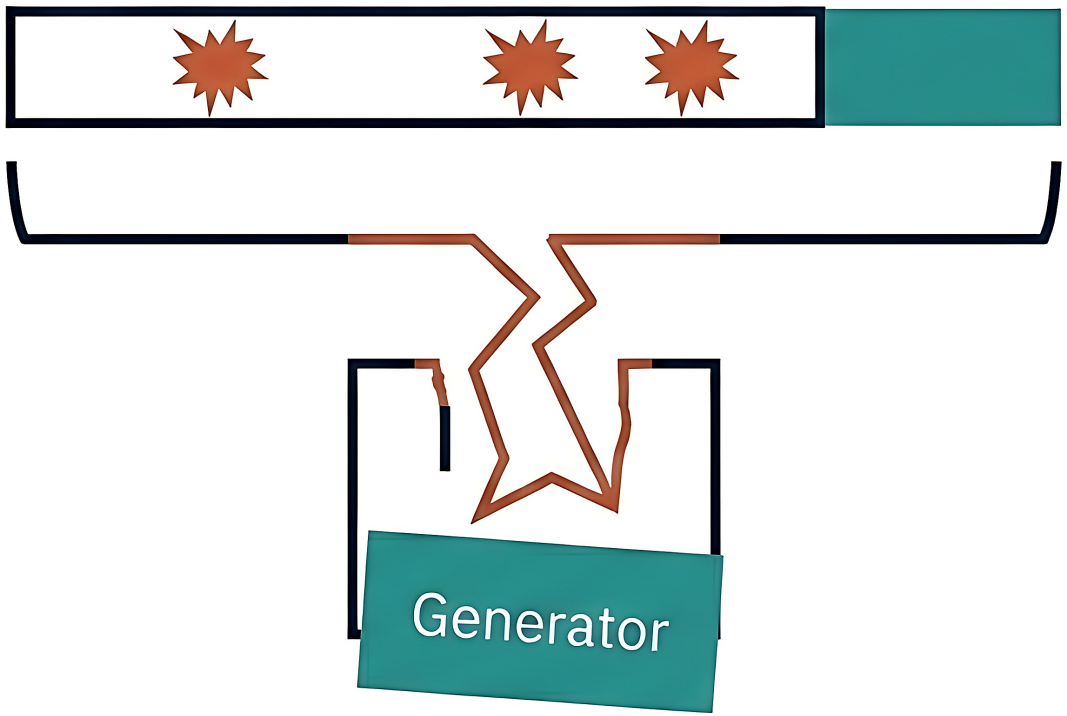
Errors Break the Structural Rule

Valid structure



Passes CRC

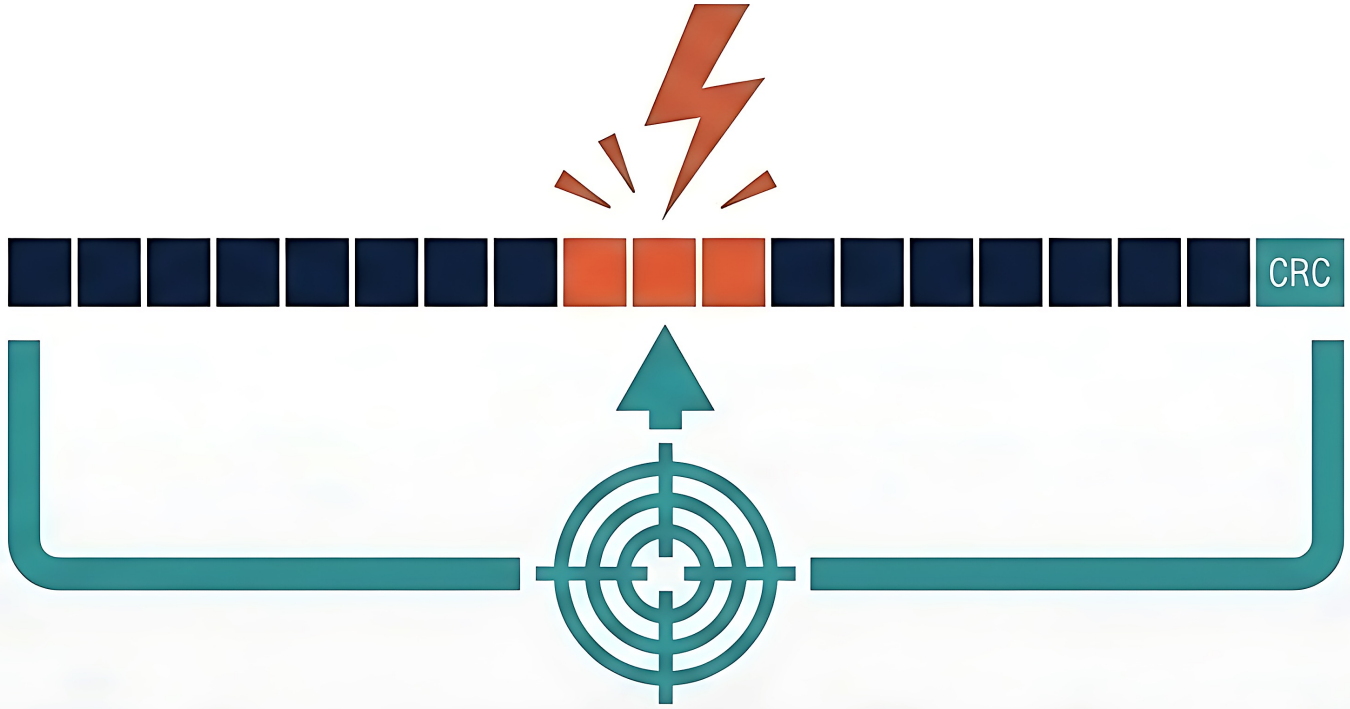
Bit flips ➔ Structure broken



CRC check fails

The Specific Threat of Burst Errors

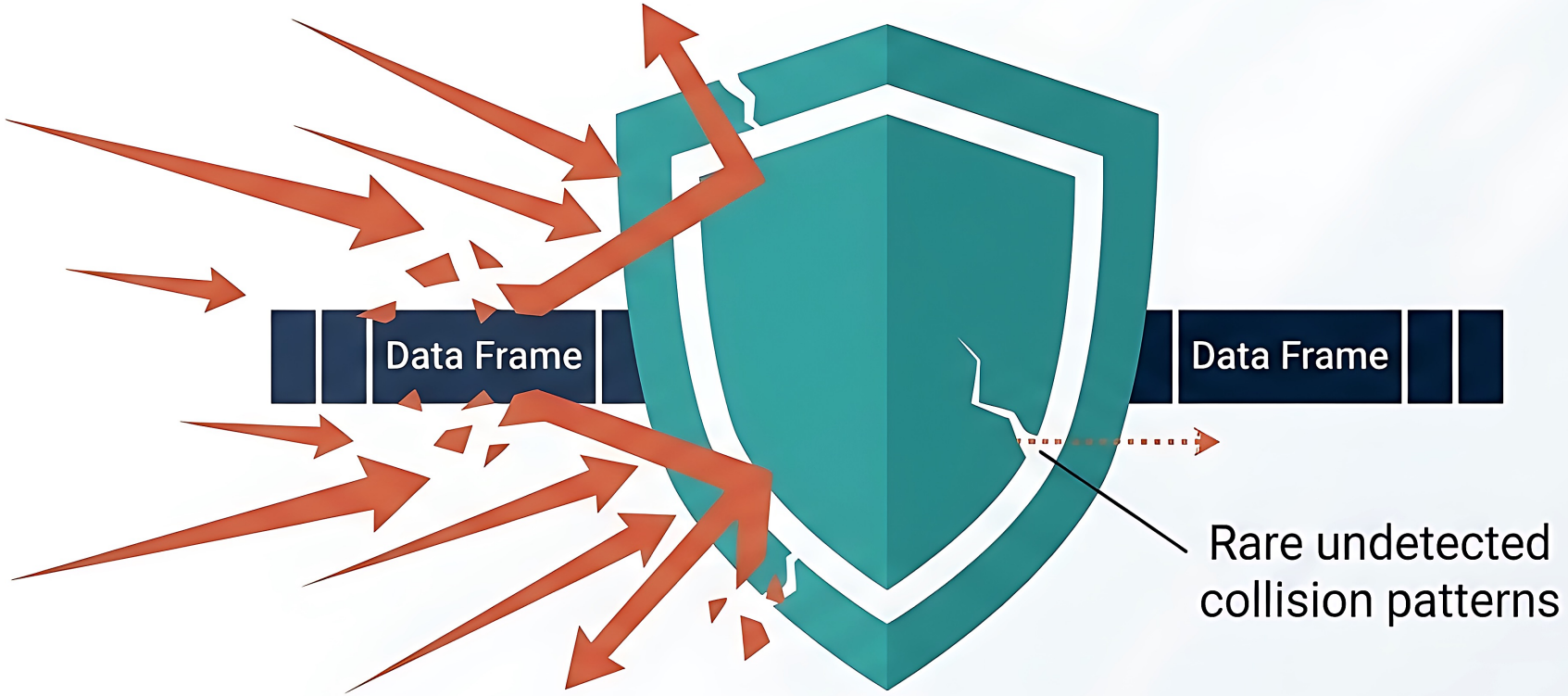
- Errors often occur in concentrated bursts due to physical link interference.
- CRC is engineered specifically to excel at detecting clustered, multi-bit corruption.



Concentrated Link-Layer Corruption

Good generator choices make CRC especially strong at detecting the burst errors common in real communication systems.

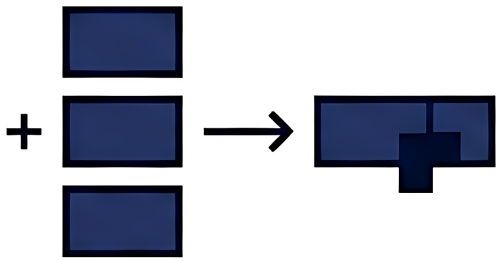
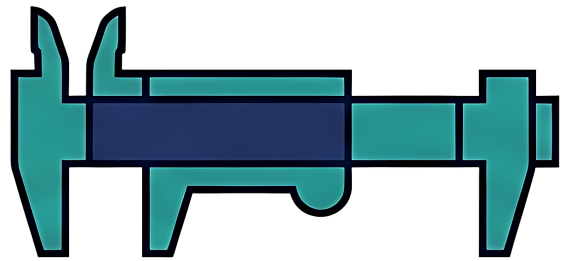
Extremely Strong, But Not Conceptually Perfect



Detects nearly all common link errors

CRC does not guarantee the detection of every imaginable error pattern. However, with a mathematically robust generator, undetected cases become exceptionally rare.

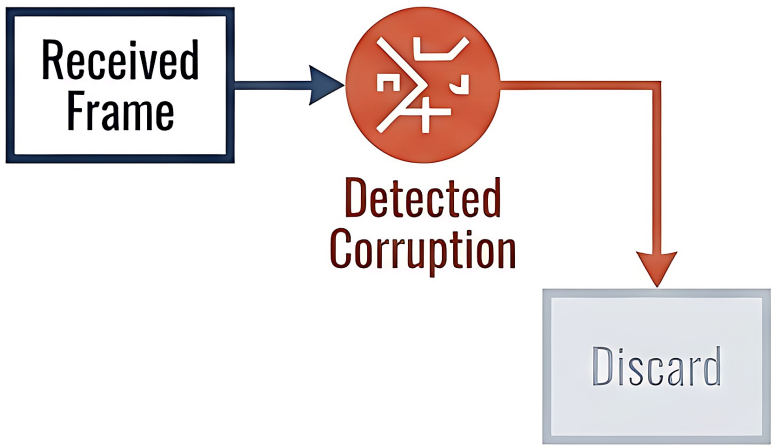
The Evolution of Redundancy

Parity	Checksum	CRC
		
Simple Rule	**Arithmetic Summary**	**Structural Enforcement**
Counts 1s	Adds chunks	Polynomial division
Weak against multi-bit errors	Better, but collisions happen	Extremely resilient to burst errors

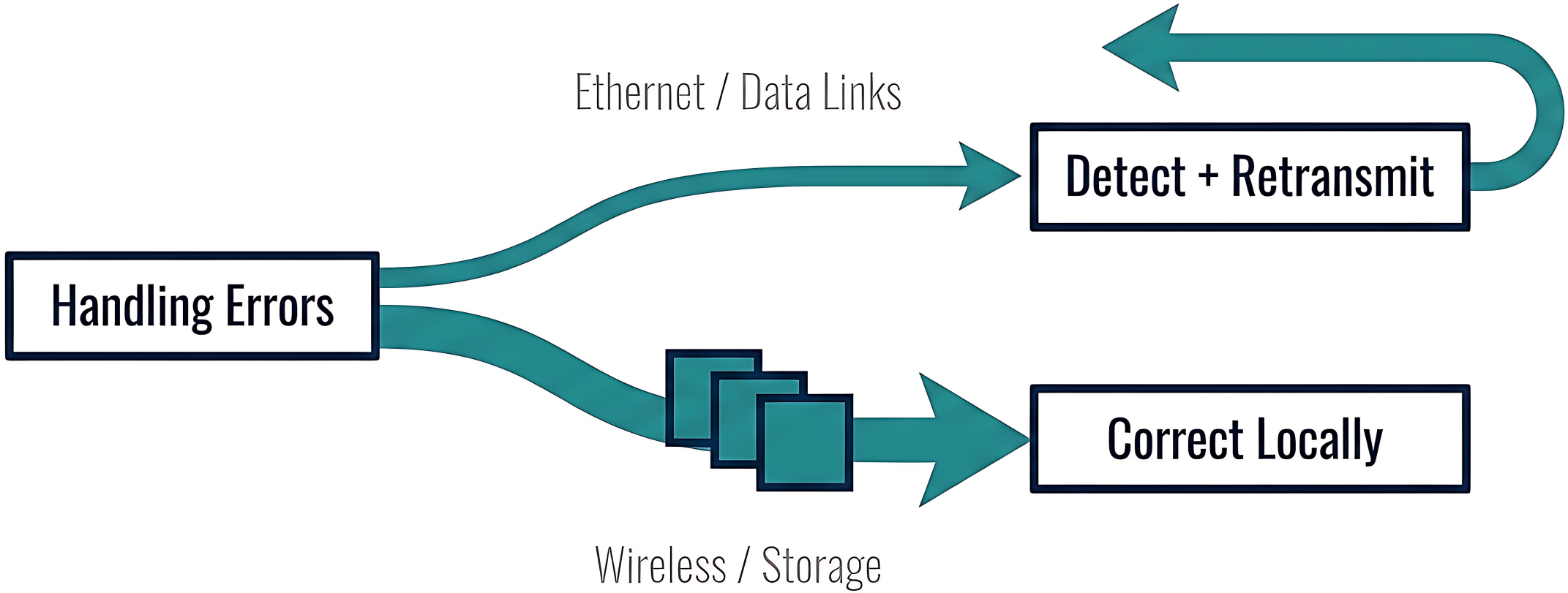
All three mechanisms add redundancy. The difference lies entirely in how deeply that redundancy structures the data.

From Detection to Correction

Detection: Know it is bad



System Design Trade-Offs

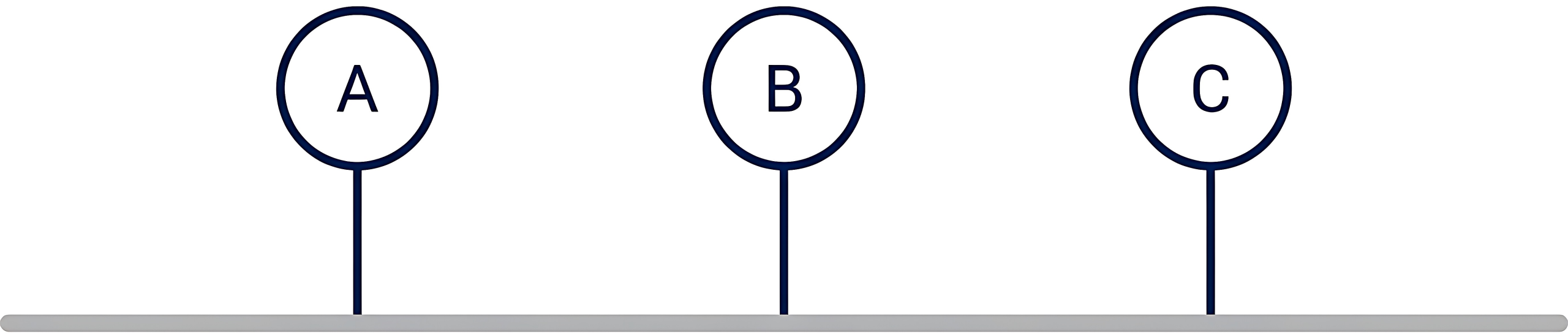


More correction usually means more overhead.

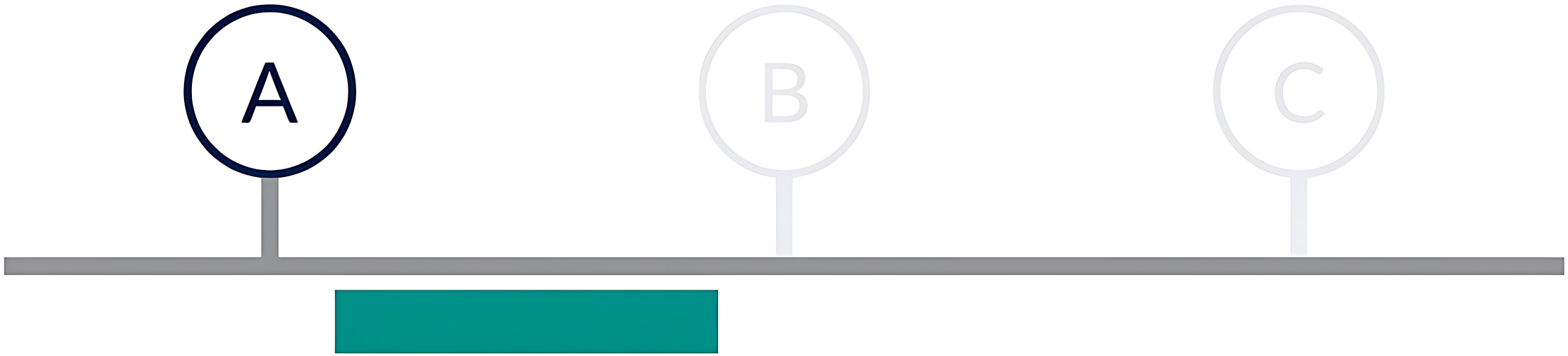
Medium Acces Control

Multiple Access Control: Sharing the Medium

How independent senders share a communication medium without destroying efficiency.



The Base Case: One Sender



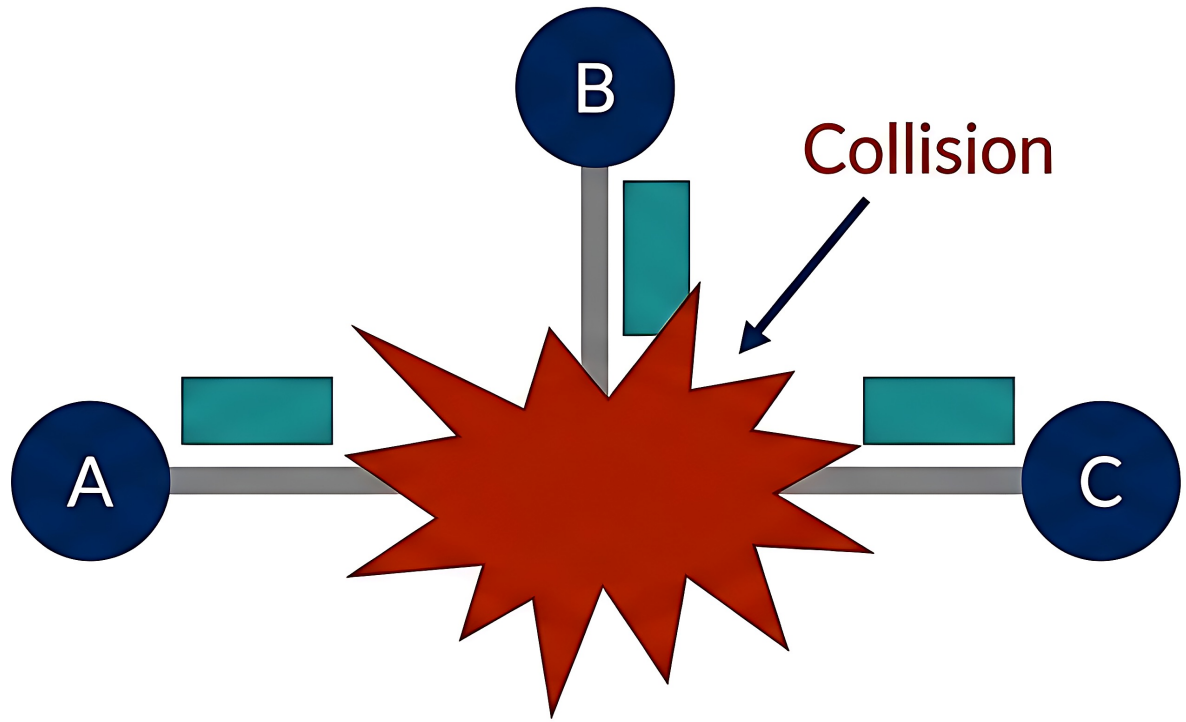
- If only one device uses the link, there is **no** access-control problem.
- No contention. No collisions. Just transmit.

The Problem: The Shared Medium

Nodes A, B, and C are independent.

When signals overlap, the receiver cannot decode them.

The real cost: Channel time and bandwidth are consumed, but zero useful frames get through.
Wasted capacity.



The Shared Wire

The Goals of a MAC Protocol

A good protocol balances competing goals:

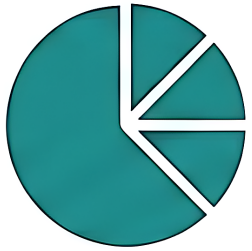
1. Efficiency: Maximize channel usage.

MAC is a tradeoff problem, constrained by the physical properties of the medium.

The Goals of a MAC Protocol

A good protocol balances competing goals:

1. Efficiency: Maximize channel usage.



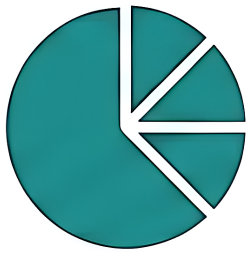
2. Fairness: Everyone gets a turn.

MAC is a tradeoff problem, constrained by the physical properties of the medium.

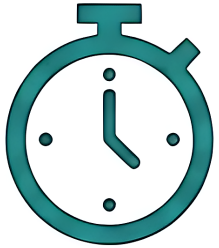
The Goals of a MAC Protocol

A good protocol balances competing goals:

1. Efficiency: Maximize channel usage.



2. Fairness: Everyone gets a turn.



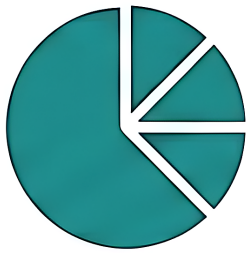
3. Delay: Minimize wait times.

MAC is a tradeoff problem, constrained by the physical properties of the medium.

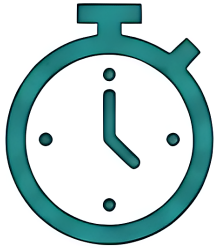
The Goals of a MAC Protocol

A good protocol balances competing goals:

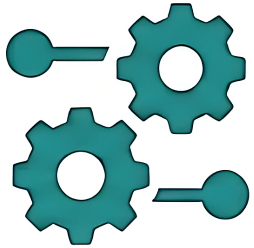
1. Efficiency: Maximize channel usage.



2. Fairness: Everyone gets a turn.



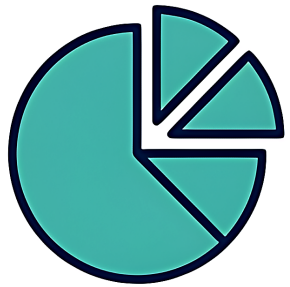
3. Delay: Minimize wait times.



4. Overhead: Keep coordination simple.

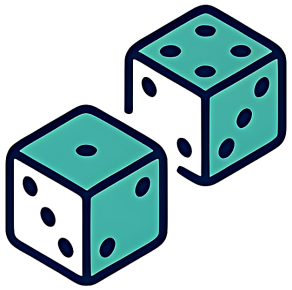
MAC is a tradeoff problem, constrained by the physical properties of the medium.

The Design Space: Three Approaches



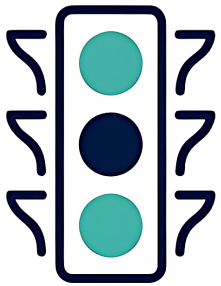
1. Partitioning

Split the resource ahead of time.



2. Random Access

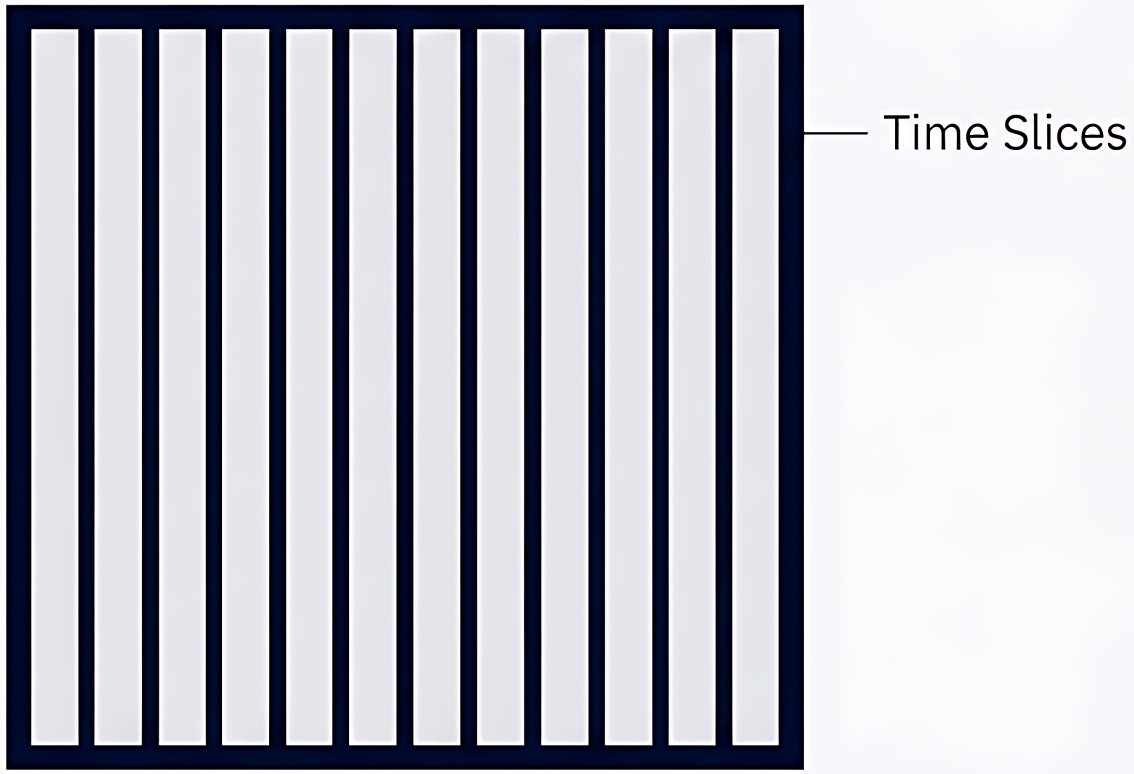
Compete only when needed.



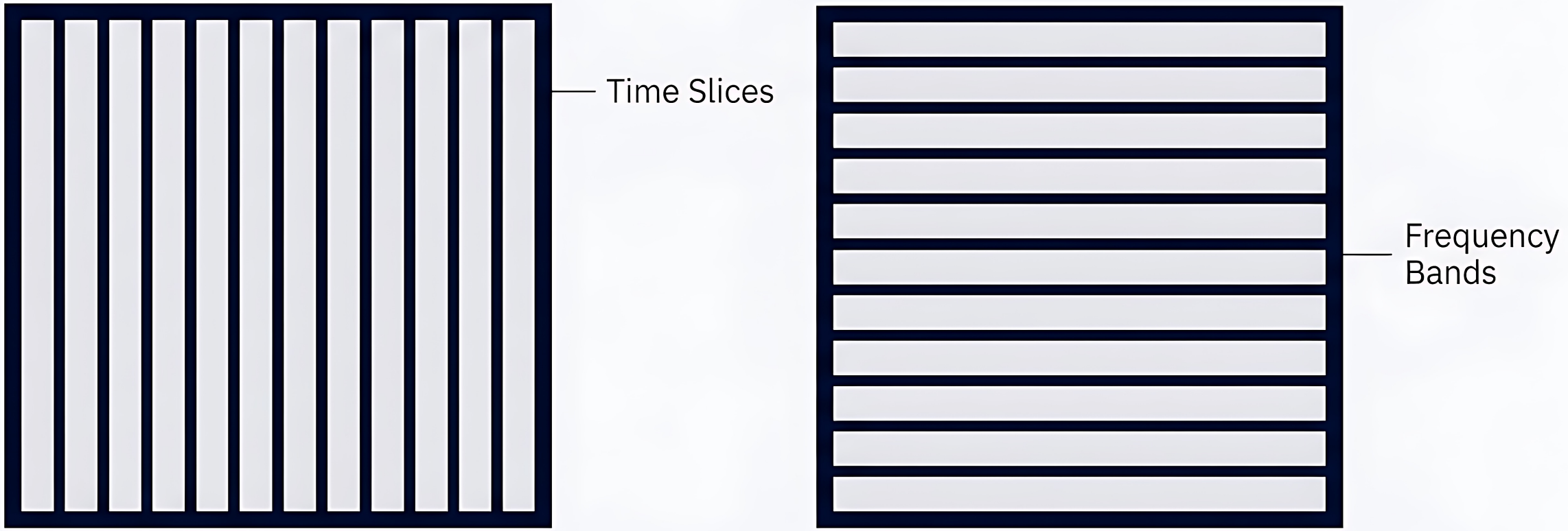
3. Taking Turns

Explicitly decide whose turn it is.

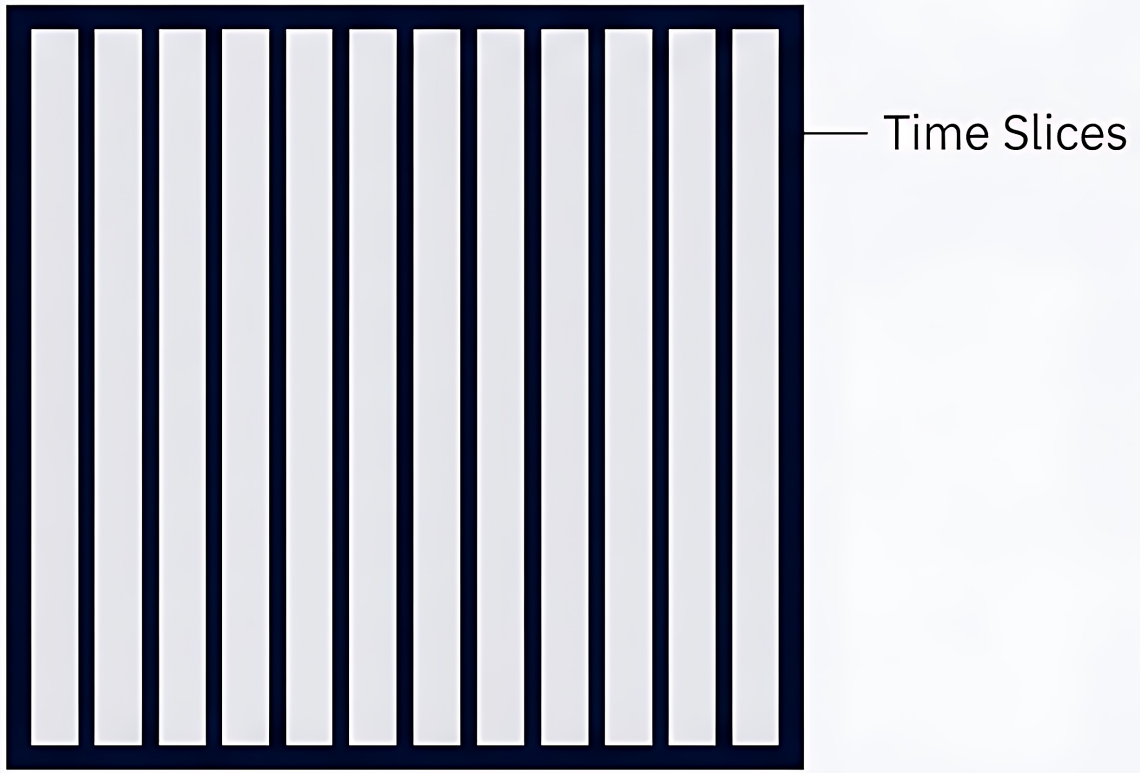
First idea: avoid conflict by partitioning the resource



First idea: avoid conflict by partitioning the resource



First idea: avoid conflict by partitioning the resource

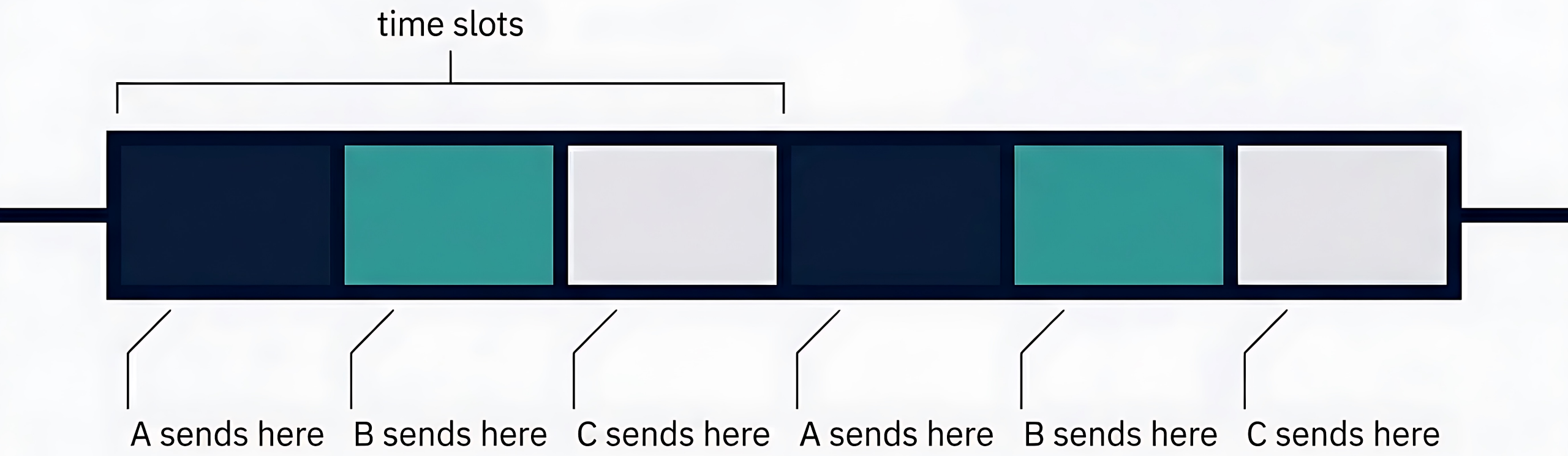


pre-assigned

no direct competition

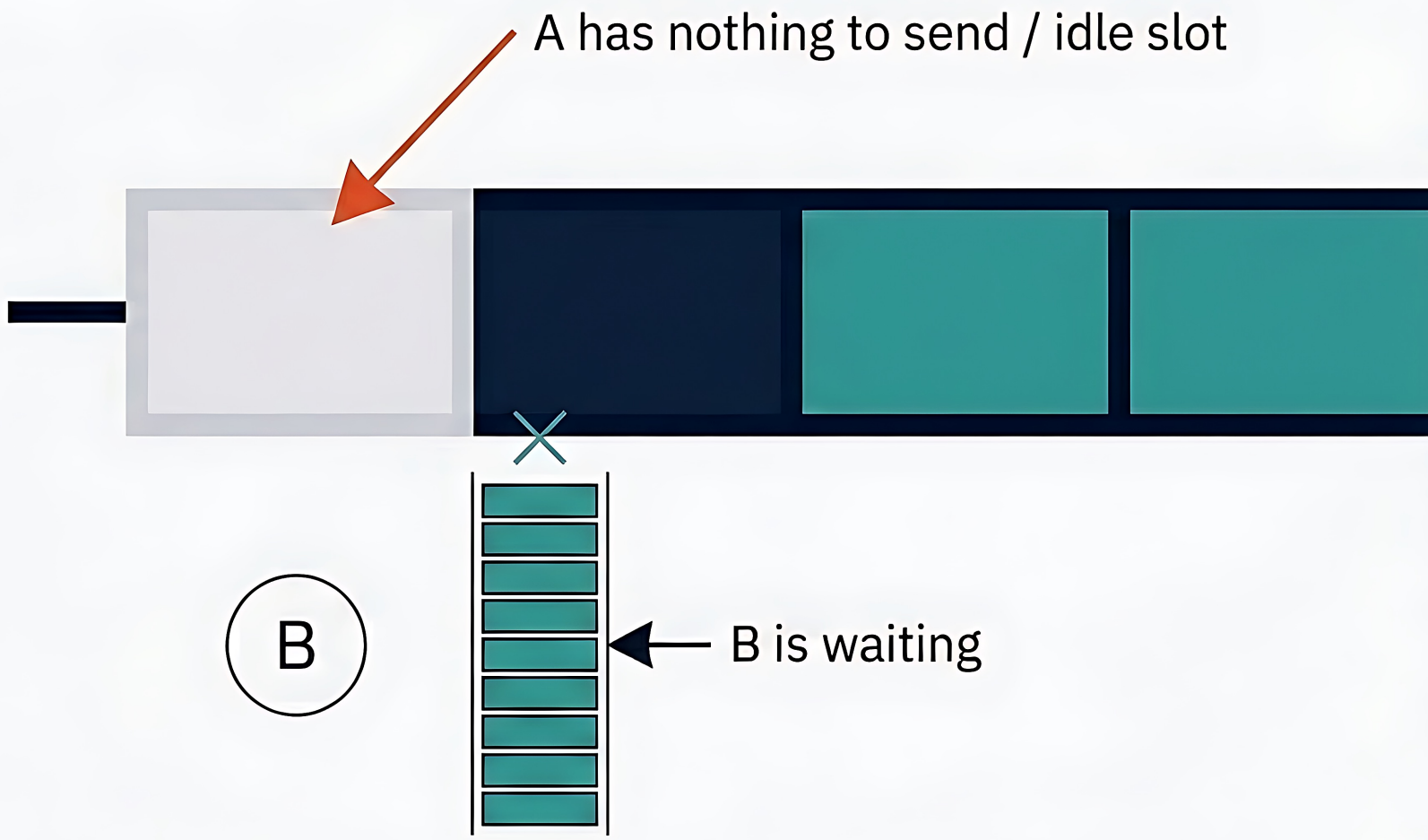
collision-free by design

TDMA separates users in time

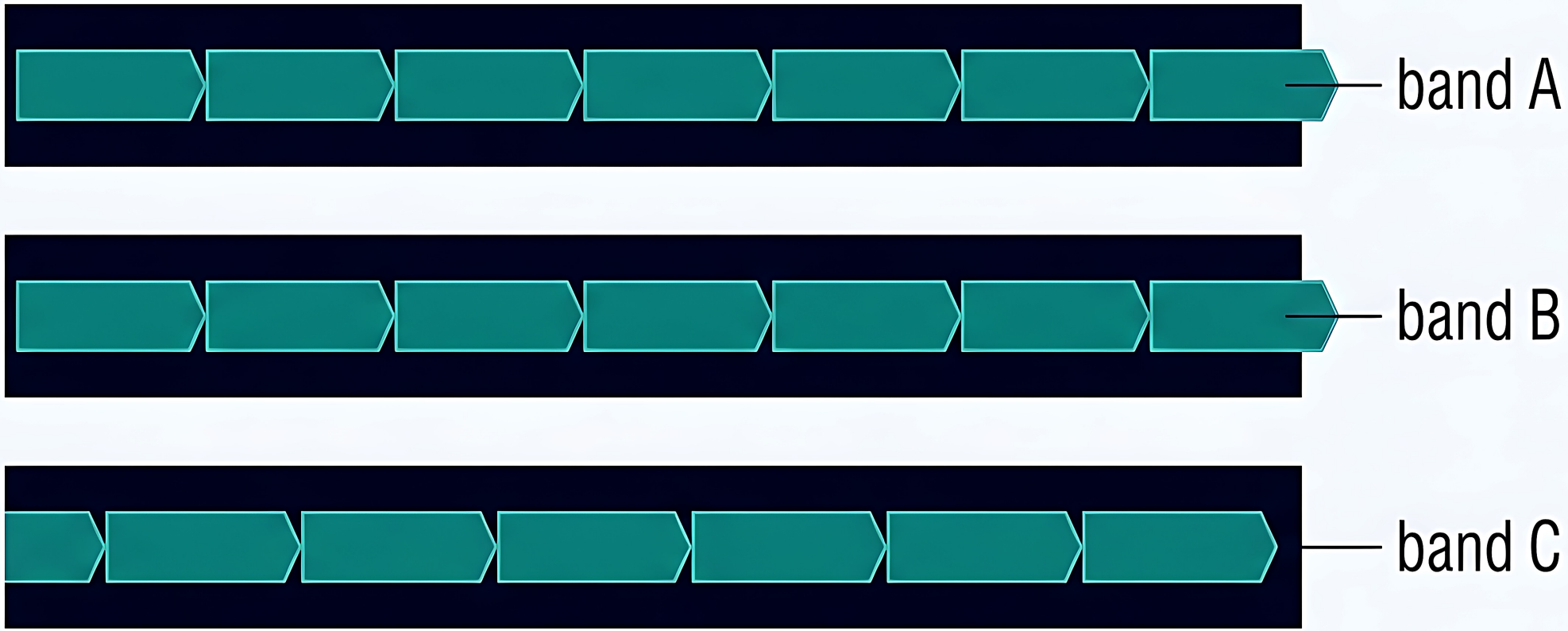


TDMA is clean, but idle slots can waste capacity

TDMA timeline slots
can waste capacity

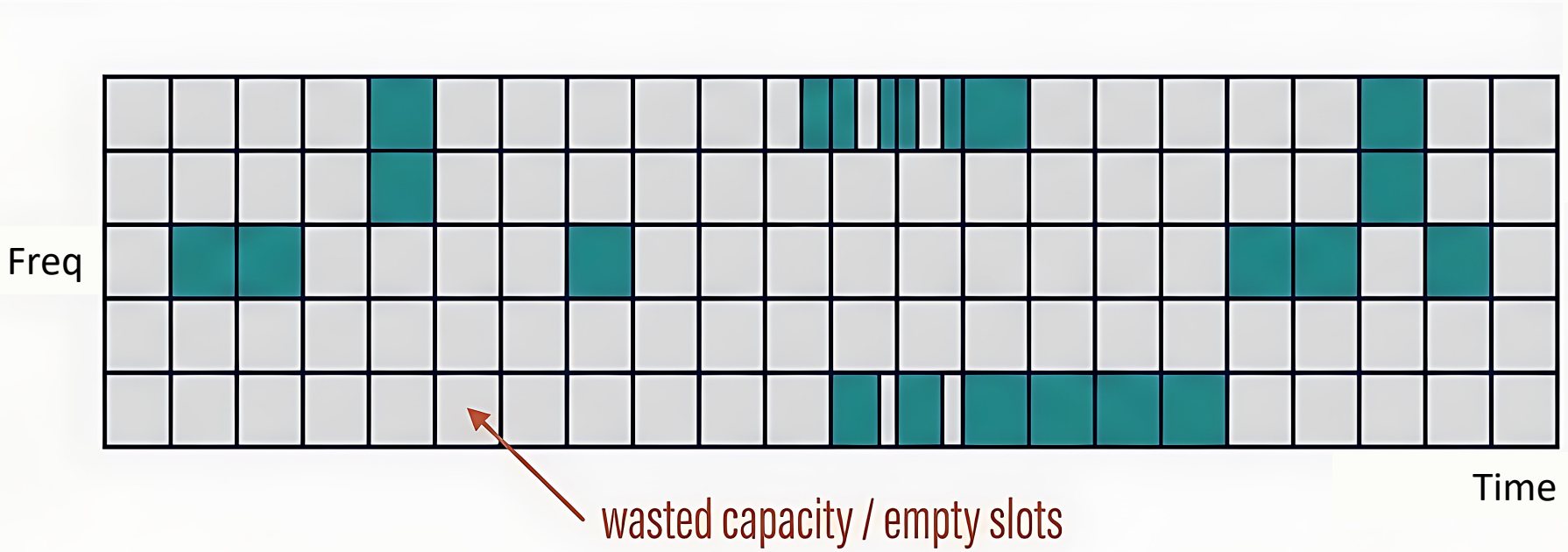


FDMA separates users in frequency

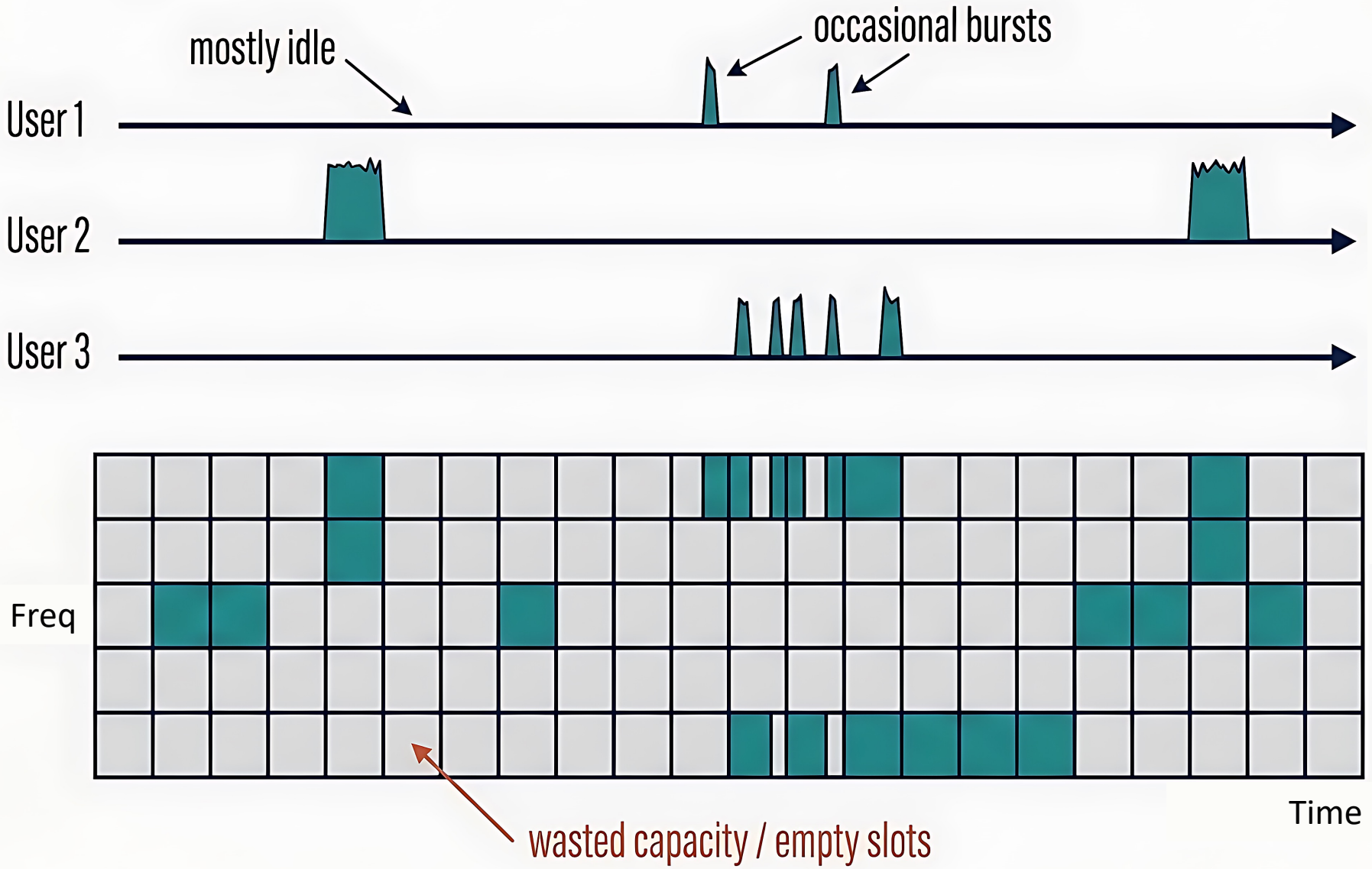


simultaneous but separated

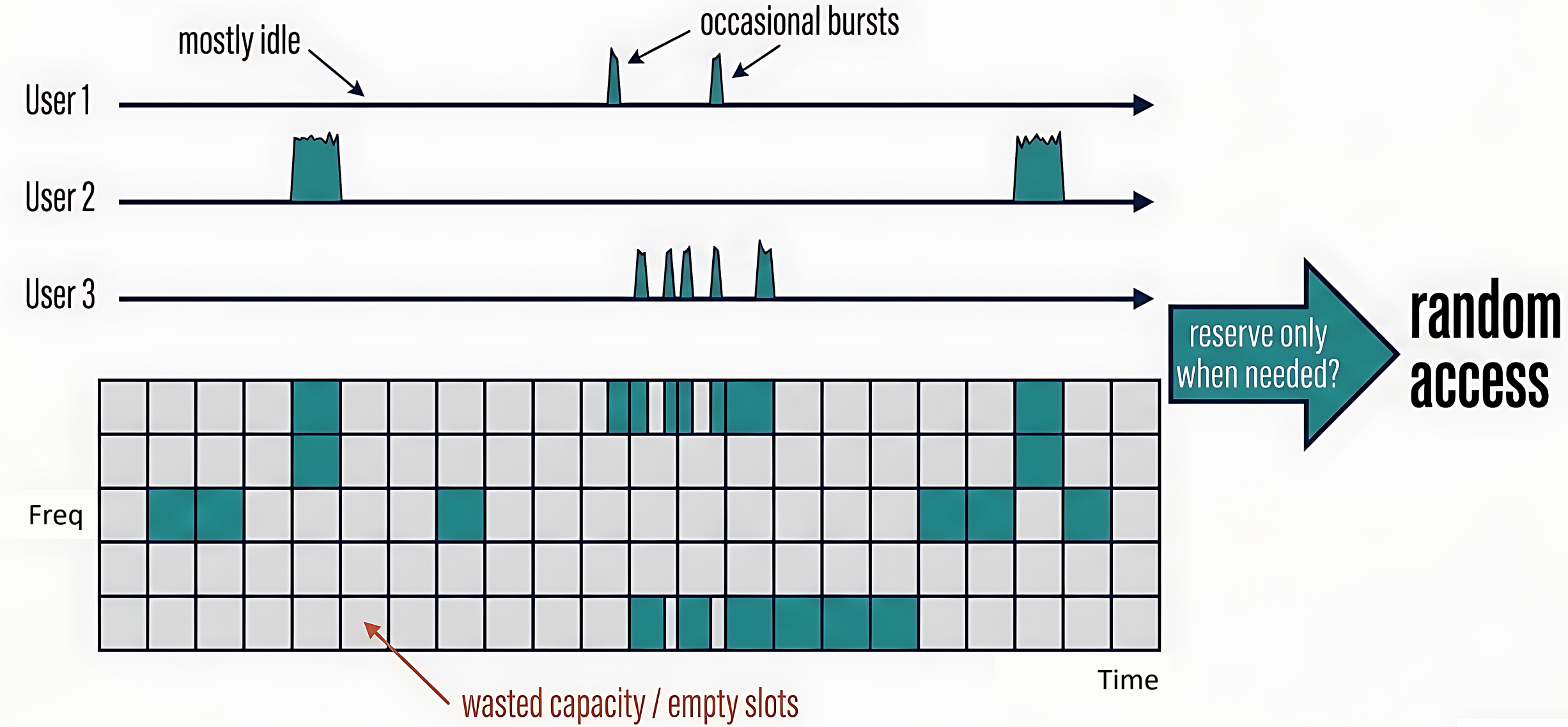
Bursty traffic makes static reservation feel wasteful



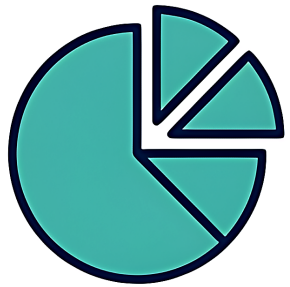
Bursty traffic makes static reservation feel wasteful



Bursty traffic makes static reservation feel wasteful

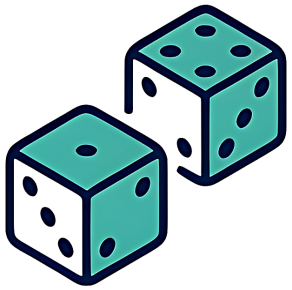


The Design Space: Three Approaches



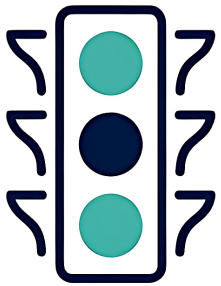
1. Partitioning

Split the resource ahead of time.



2. Random Access

Compete only when needed.



3. Taking Turns

Explicitly decide whose turn it is.

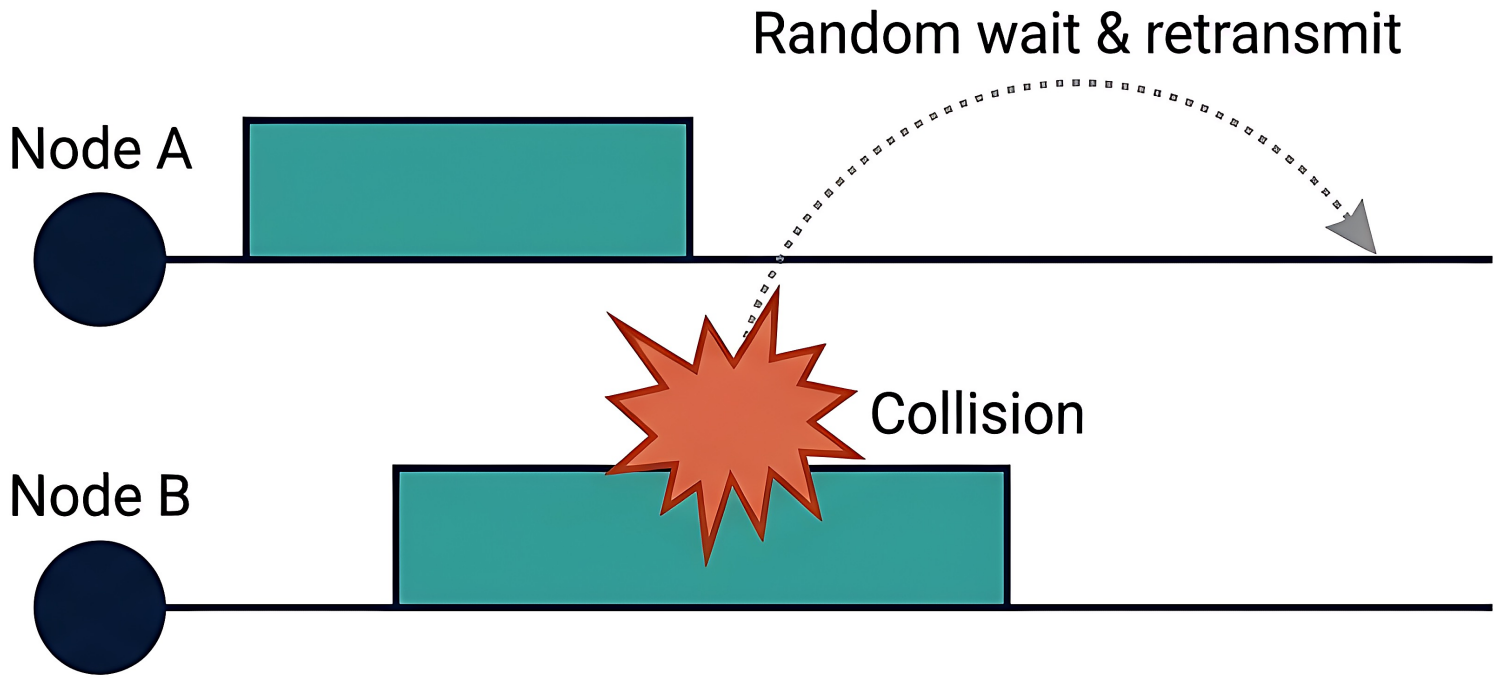
A Very Simple Idea: ALOHA

Send first, recover later: the earliest random access protocol.

No Central Scheduler: Fully distributed, highly simple operation.

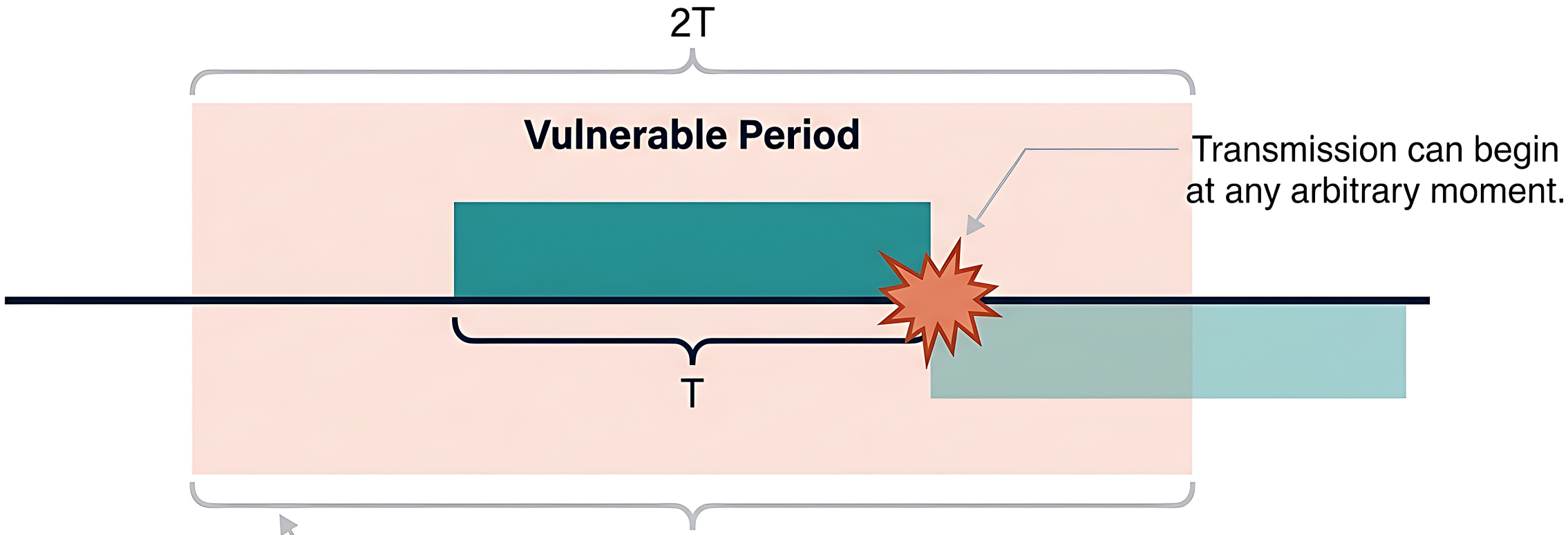
Immediate Transmission: Nodes send frames the moment data is available.

Collision Recovery: If frames overlap, nodes wait a randomized duration before trying again.



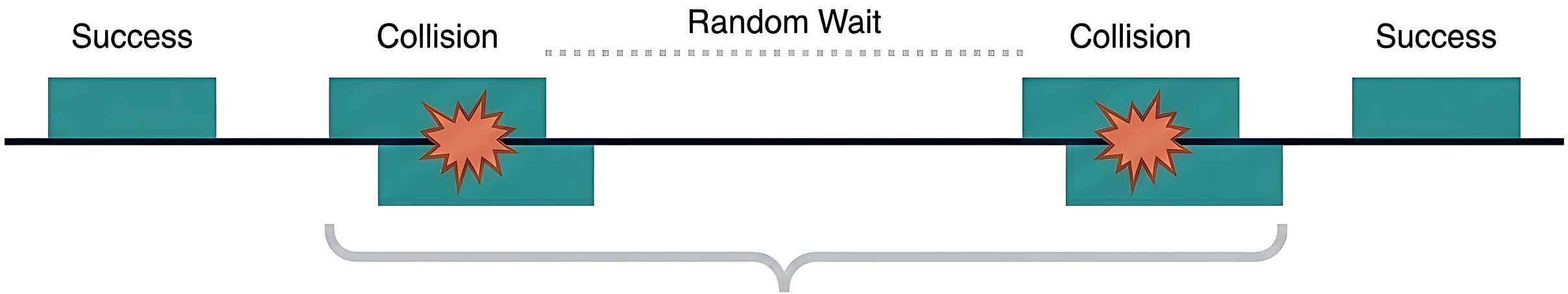
How Collisions Happen in Pure ALOHA

Uncoordinated transmission means even a tiny overlap destroys the frame.



Why Pure ALOHA is Inefficient

The channel spends most of its time carrying doomed frames.



Arbitrary Start Times:
Collisions happen far too easily without structural alignment.

Compounding Waste:
A single collision forces multiple retransmissions, severely clogging the channel.

Maximum Throughput:
Only **~18%** under heavy load.

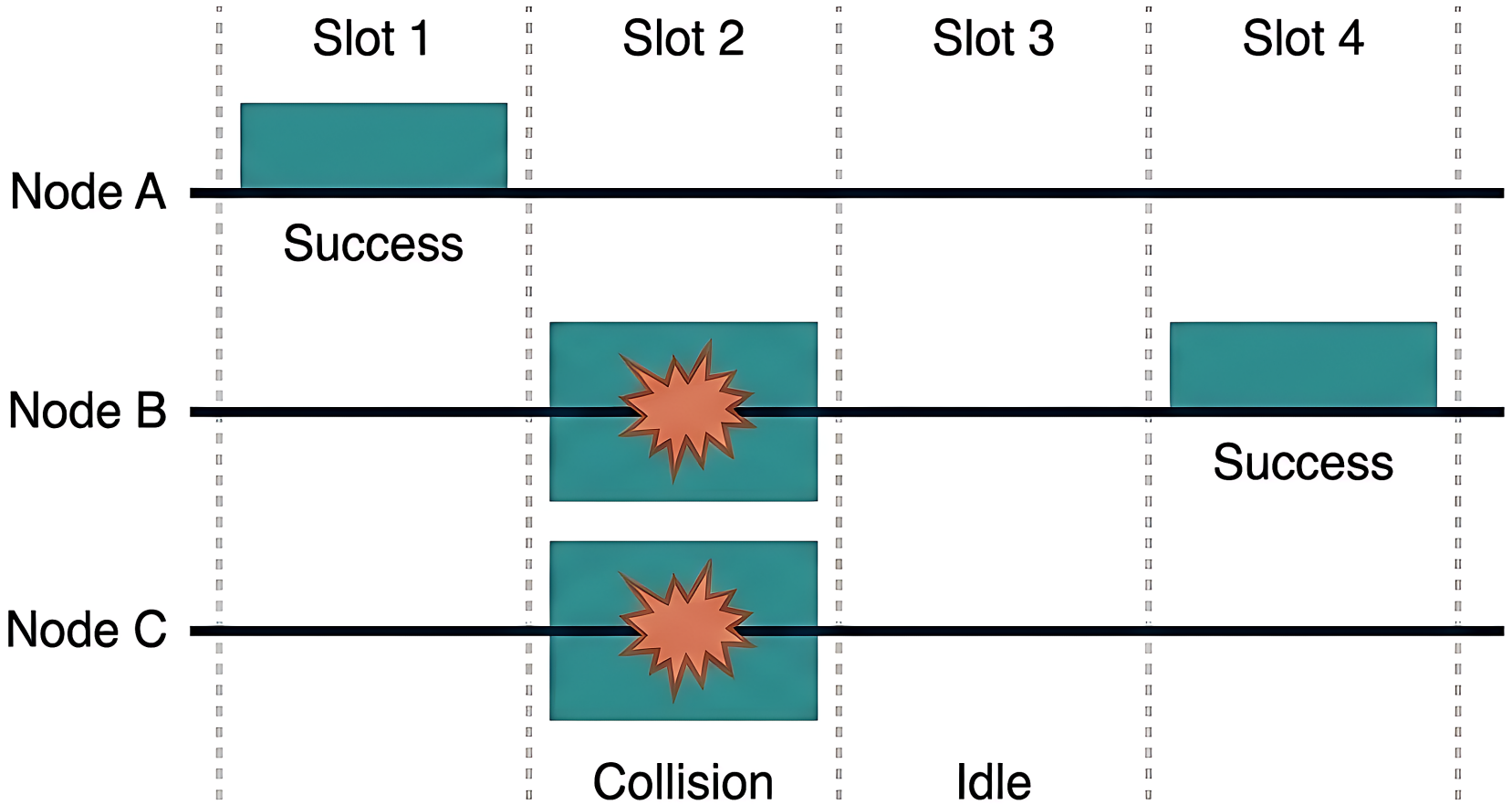
Improvement: Slotted ALOHA

Adding a simple constraint—time slots—drastically reduces collisions.

Aligned Time:
Time is divided into equal, discrete slots.

Strict Boundaries:
Nodes must wait for the beginning of the next slot to transmit.

Eliminates Partial Overlap:
Collisions now only happen if nodes pick the exact same slot.



Pure ALOHA vs. Slotted ALOHA

A small amount of coordination significantly improves system performance.

Pure ALOHA

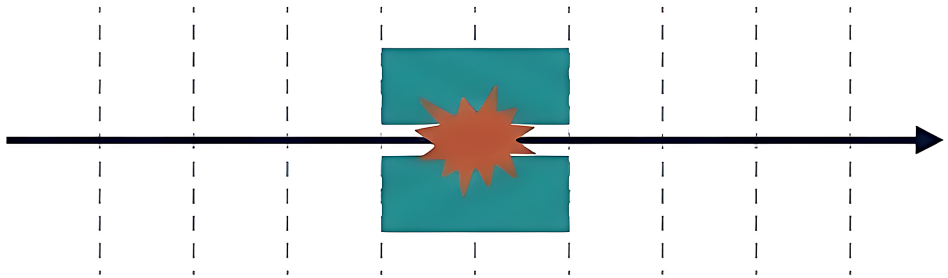
Send Rules: Anytime, completely uncoordinated.

Complexity: Very simple, no synchronization needed.

Failure Mode: Partial overlaps destroy frames easily.

Efficiency: ~18% max throughput.

Slotted ALOHA



Send Rules: Only at strict slot boundaries.

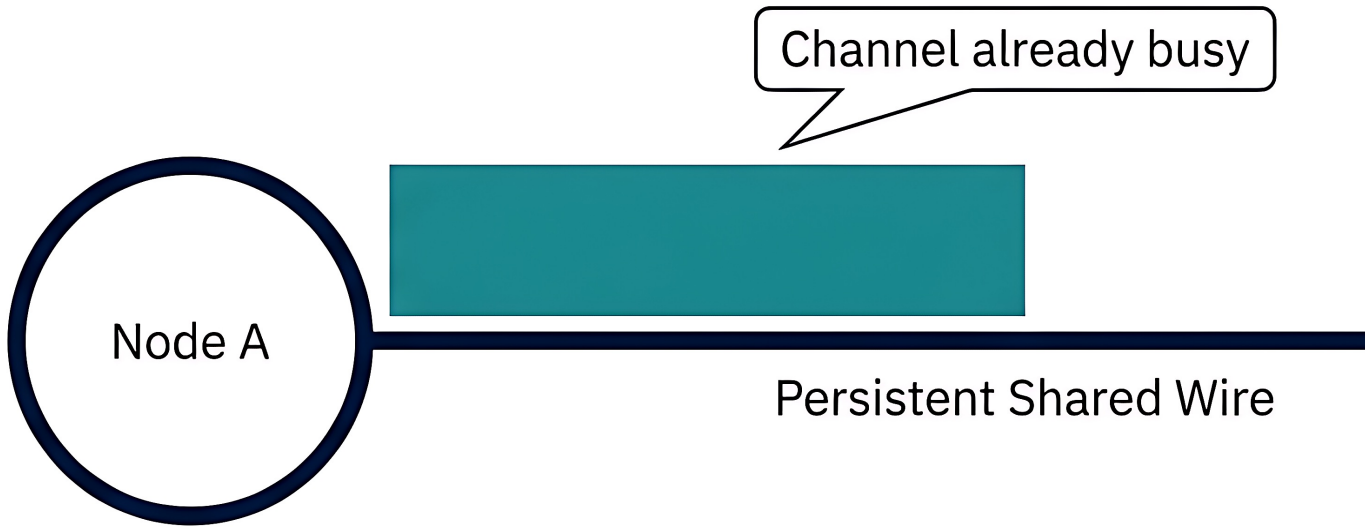
Complexity: Requires global time synchronization across nodes.

Failure Mode: Collisions restricted to full-slot overlap only.

Efficiency: **~37%** max throughput.

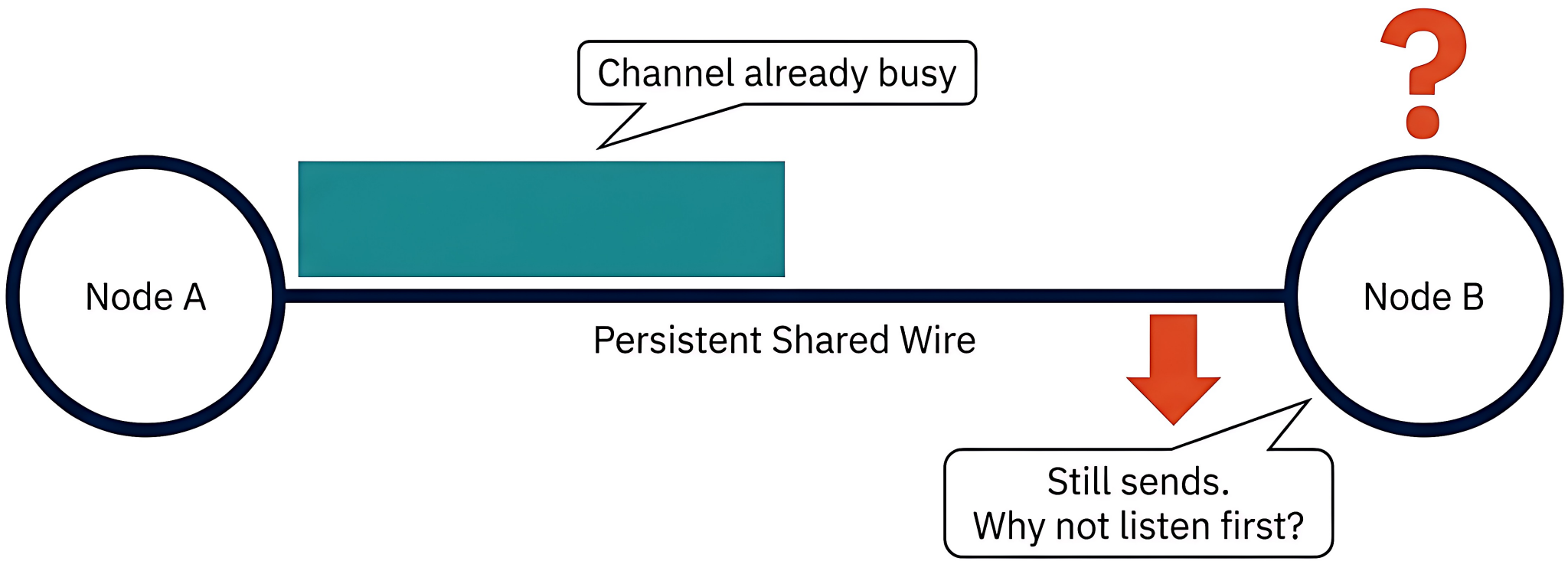
ALOHA still sends blindly

Even after slotting, ALOHA ignores one obvious clue: whether the channel is already busy.



ALOHA still sends blindly

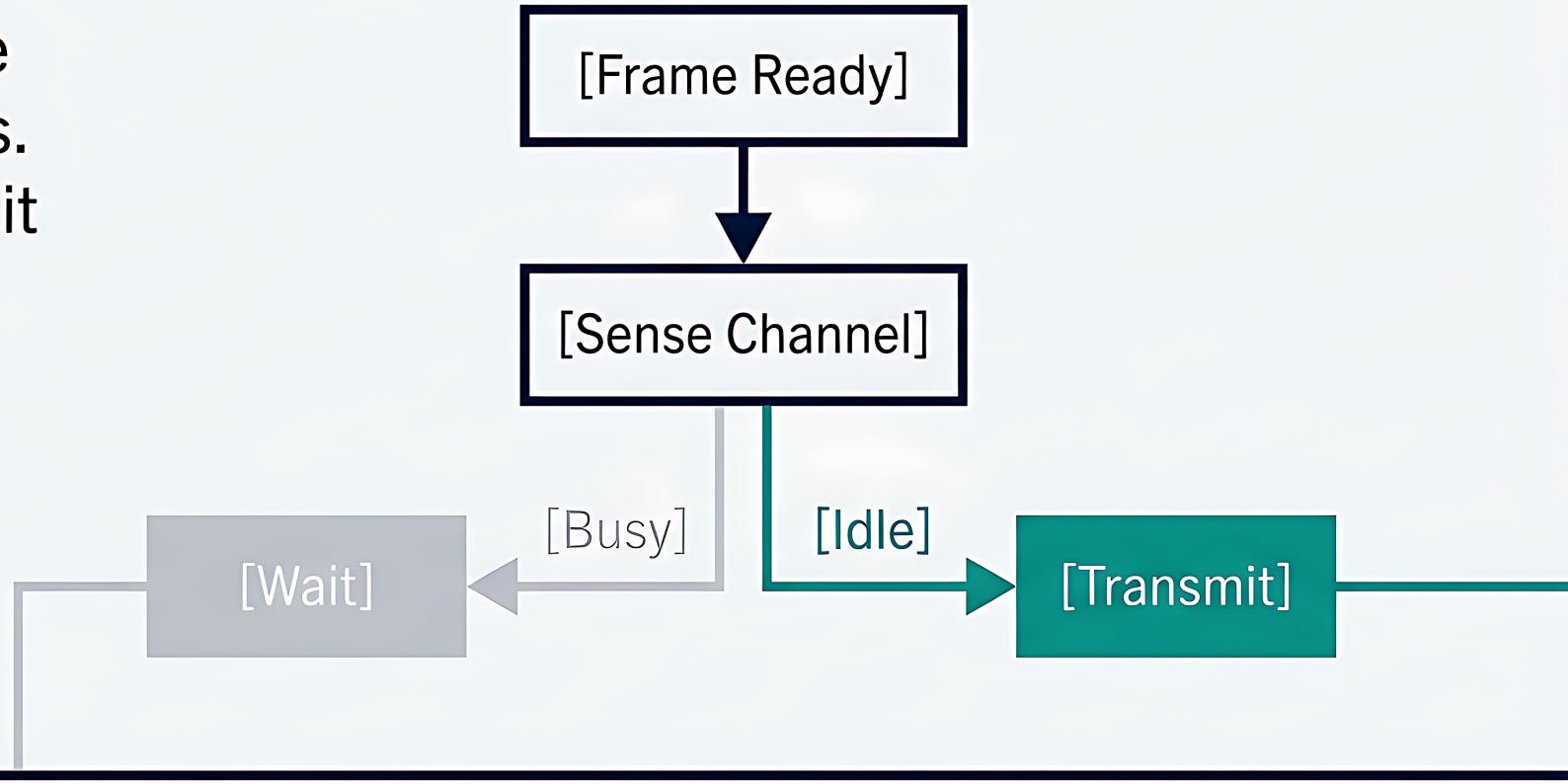
Even after slotting, ALOHA ignores one obvious clue: whether the channel is already busy.



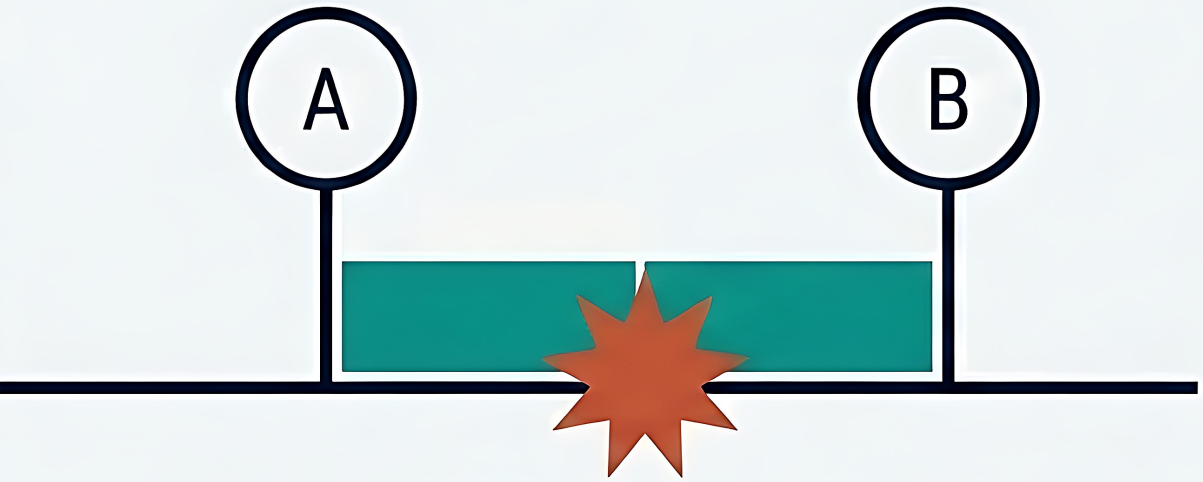
CSMA listens before transmitting

CSMA (Carrier Sense Multiple Access) adds local awareness. If the channel is busy, wait. If it appears idle, transmit.

CSMA is still random access, but smarter.

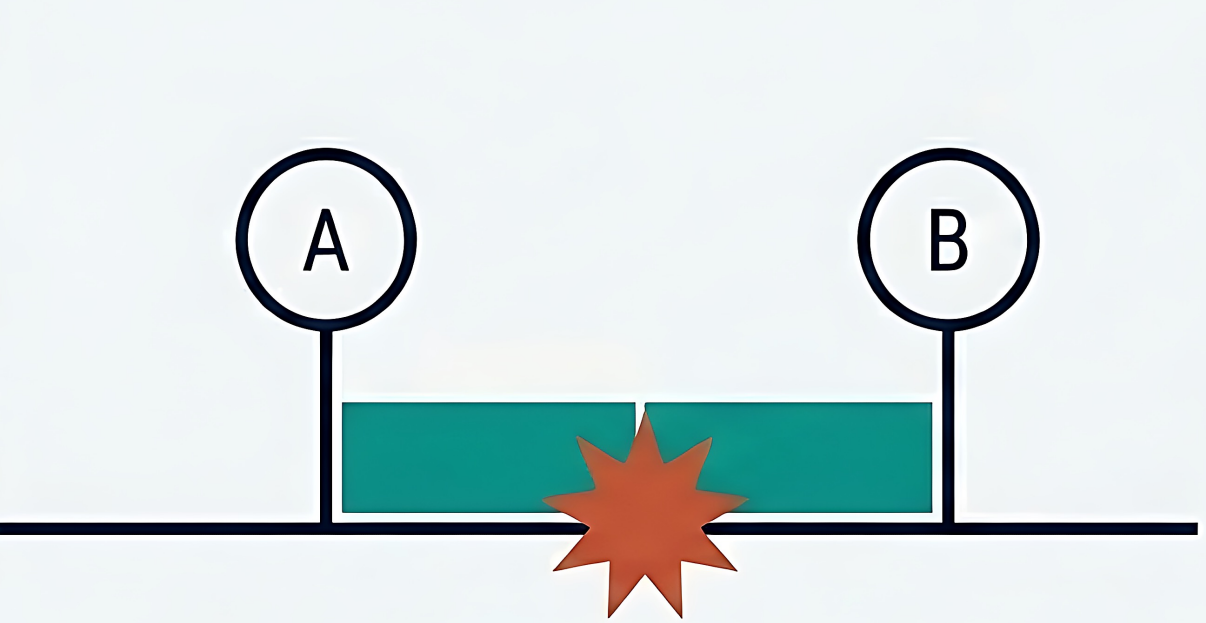


CSMA is much smarter than ALOHA

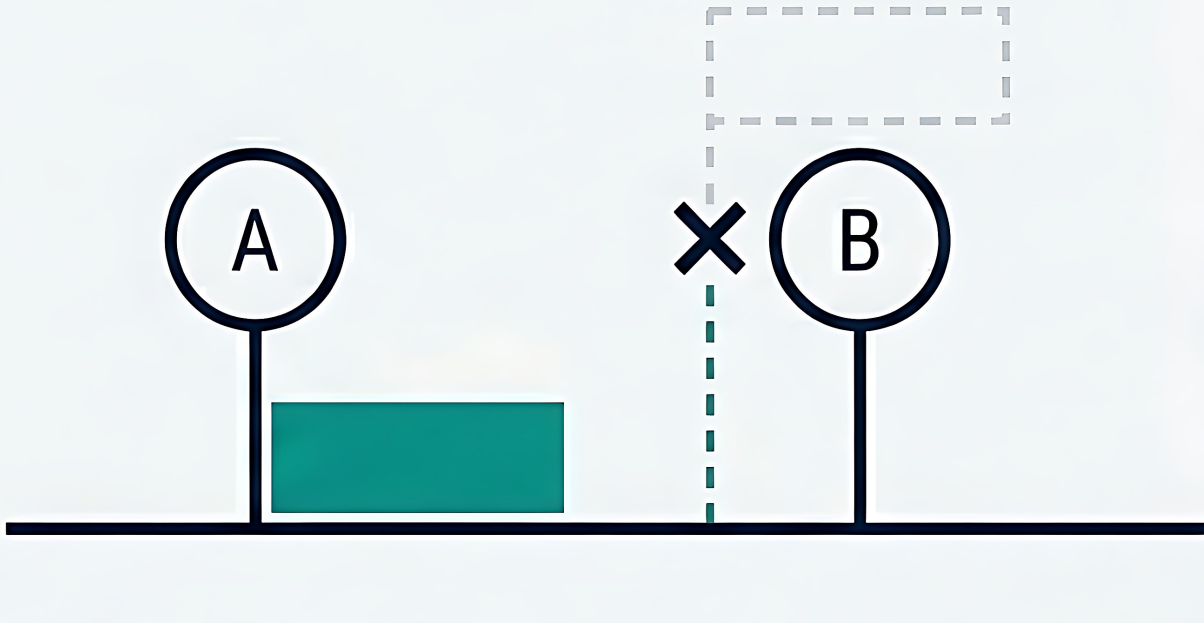


ALOHA: Send anyway

CSMA is much smarter than ALOHA



ALOHA: Send anyway



CSMA: Defer

Same situation, smarter decision.

But CSMA still cannot eliminate collisions

Carrier sensing helps, but propagation delay creates a physical blind spot.

1. A senses idle

But CSMA still cannot eliminate collisions

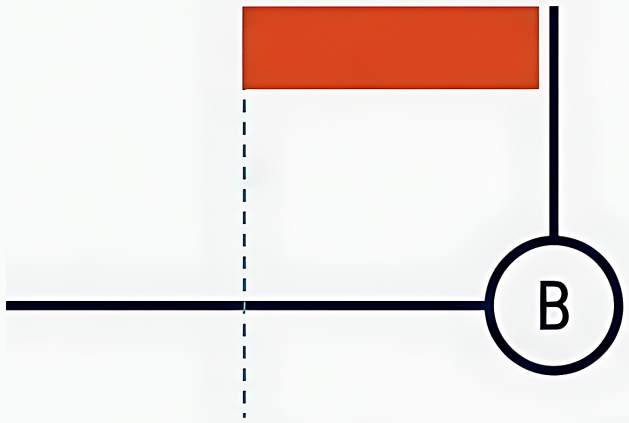
Carrier sensing helps, but propagation delay creates a physical blind spot.

1. A senses idle
2. A starts

But CSMA still cannot eliminate collisions

Carrier sensing helps, but propagation delay creates a physical blind spot.

1. A senses idle
2. A starts

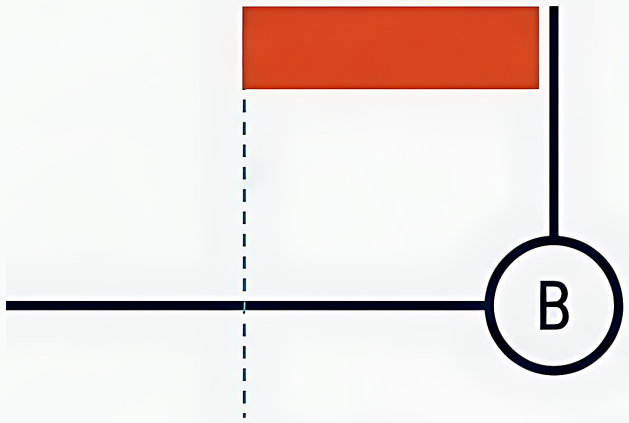


3. B still senses idle

But CSMA still cannot eliminate collisions

Carrier sensing helps, but propagation delay creates a physical blind spot.

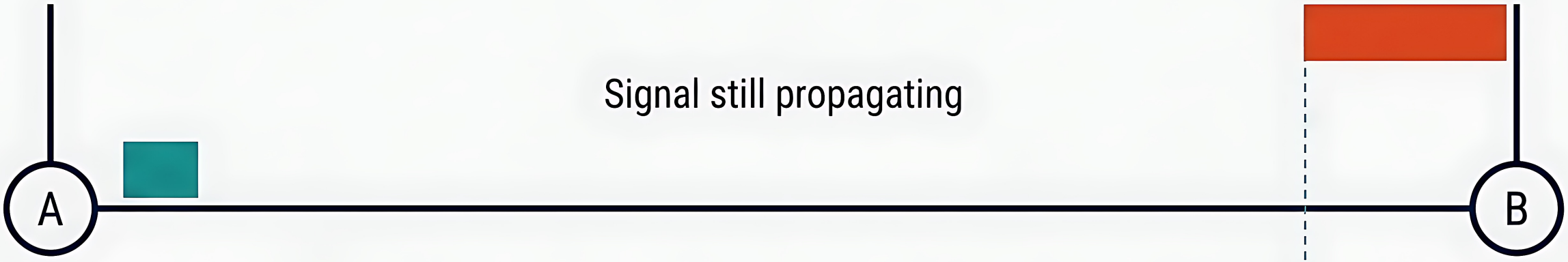
1. A senses idle
2. A starts



3. B still senses idle
4. B starts too

But CSMA still cannot eliminate collisions

Carrier sensing helps, but propagation delay creates a physical blind spot.

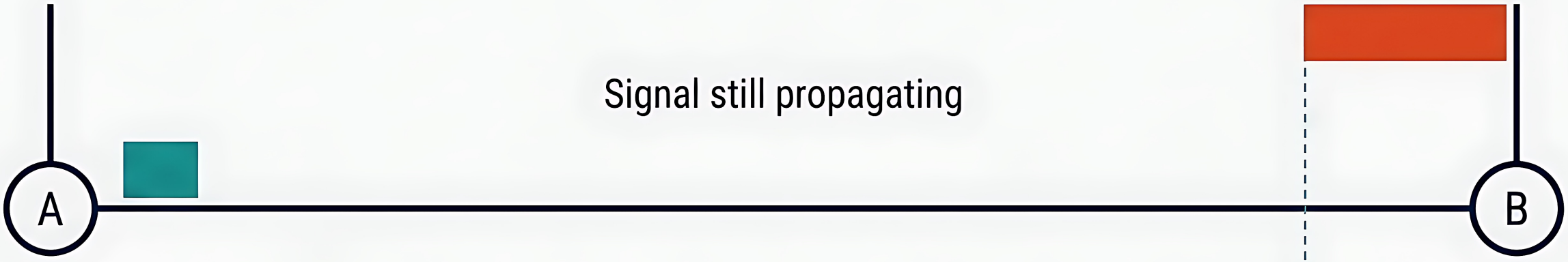


1. A senses idle
2. A starts

3. B still senses idle
4. B starts too

But CSMA still cannot eliminate collisions

Carrier sensing helps, but propagation delay creates a physical blind spot.



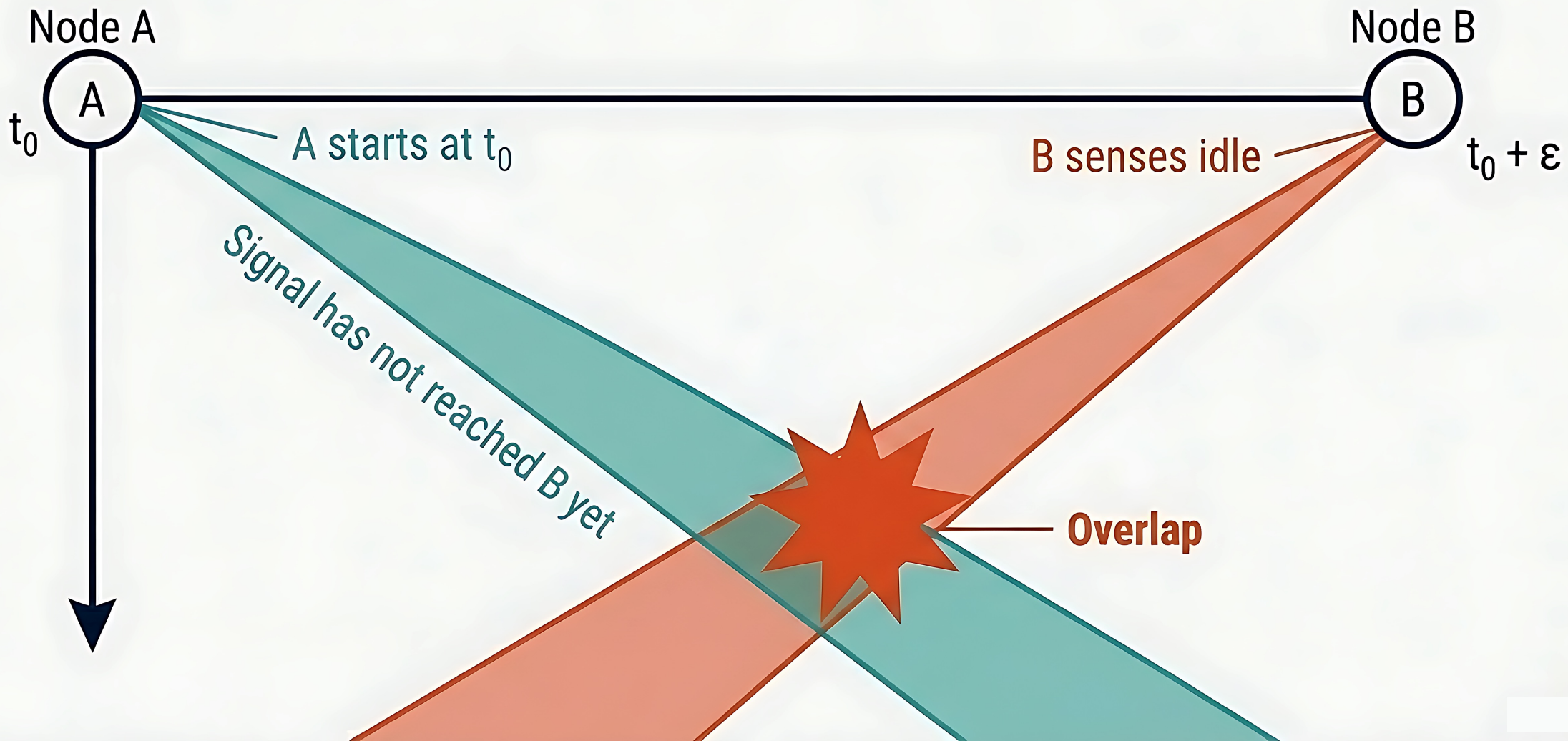
1. A senses idle
2. A starts

3. B still senses idle
4. B starts too

→ **Collision**

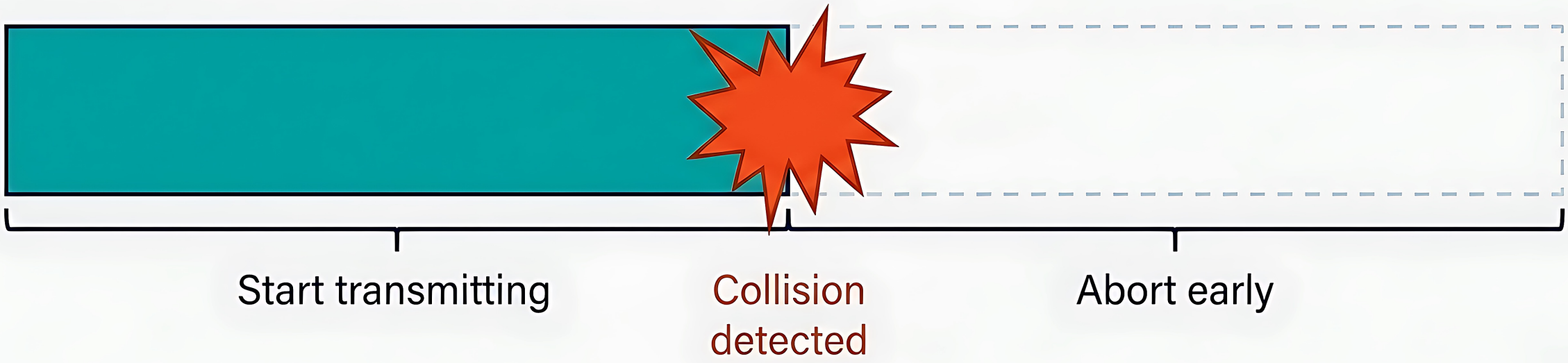
"Idle here" does not mean "idle everywhere"

The medium is physical, so channel state is local and delayed.



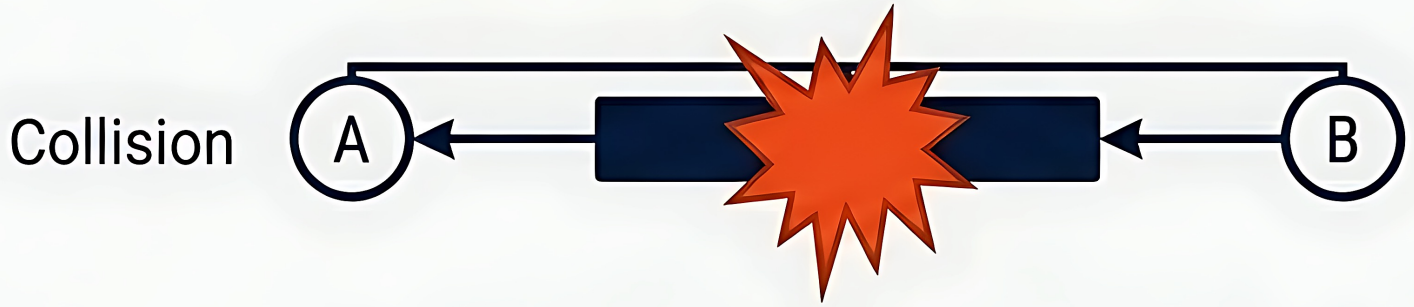
CSMA/CD cuts the loss short when a collision occurs

If a collision is unavoidable, at least do not waste the entire frame.



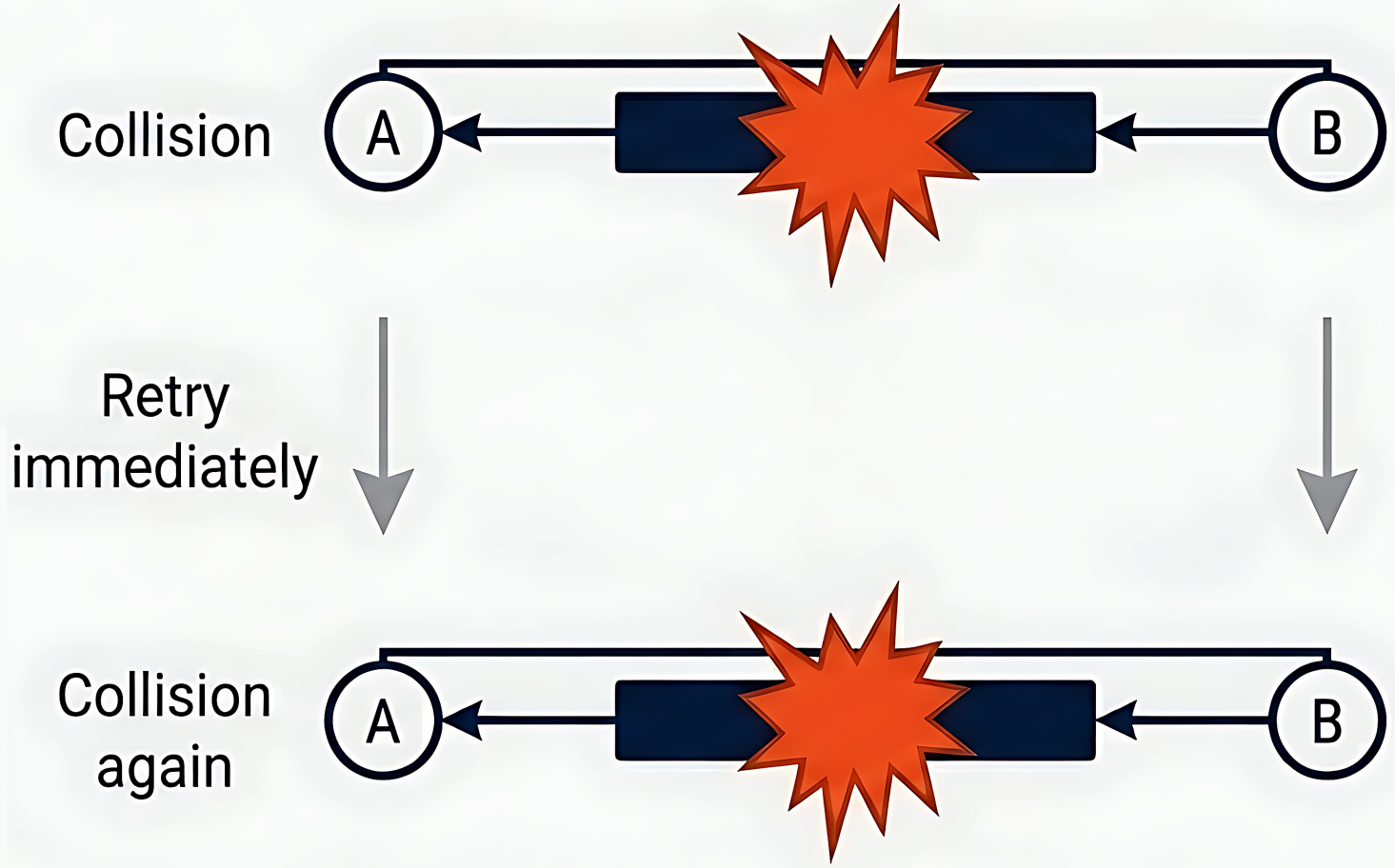
After a collision, immediate retry causes repeated collisions

If the same two nodes collide and both retry immediately in the exact same way, they will just collide again.



After a collision, immediate retry causes repeated collisions

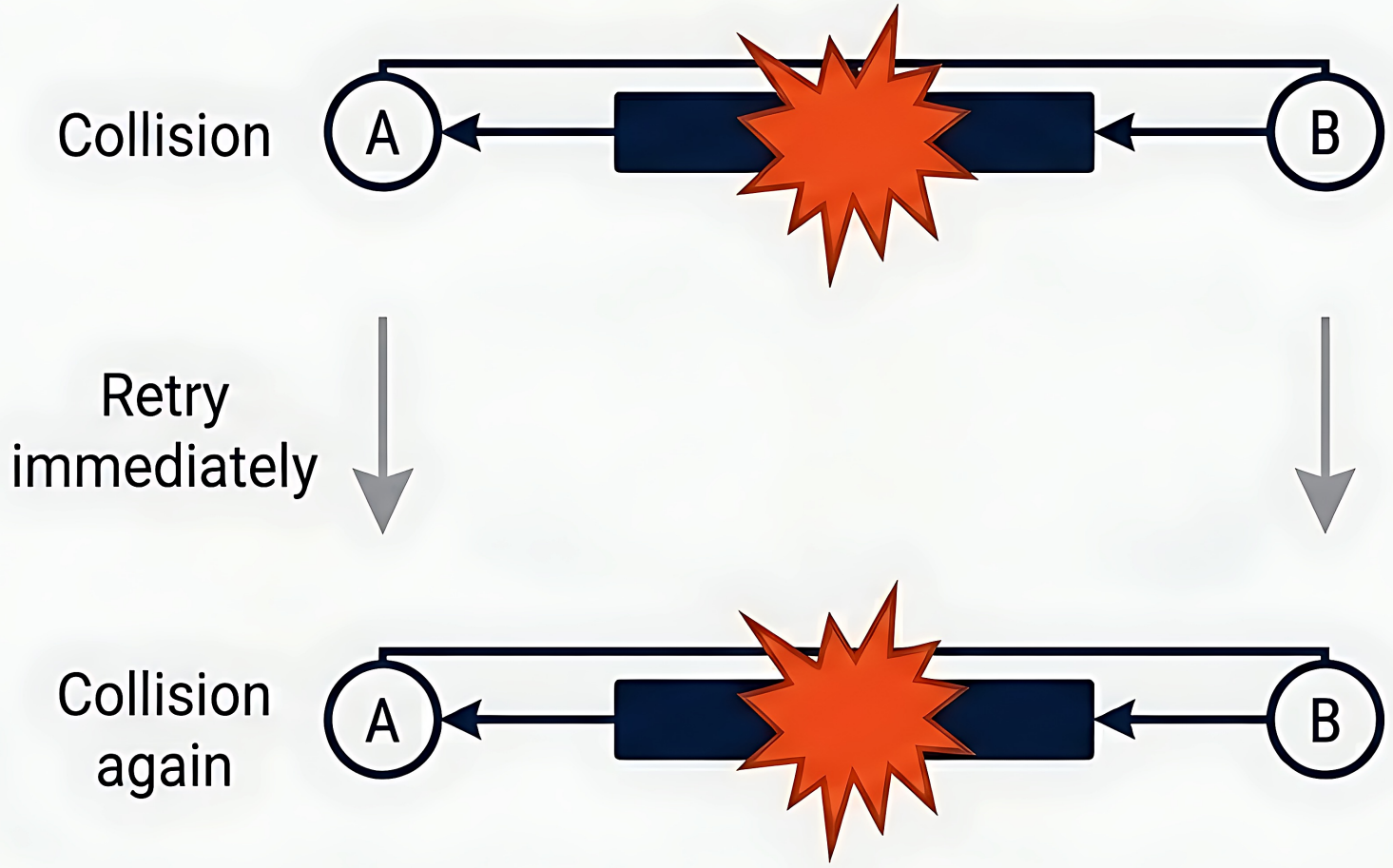
If the same two nodes collide and both retry immediately in the exact same way, they will just collide again.



After a collision, immediate retry causes repeated collisions

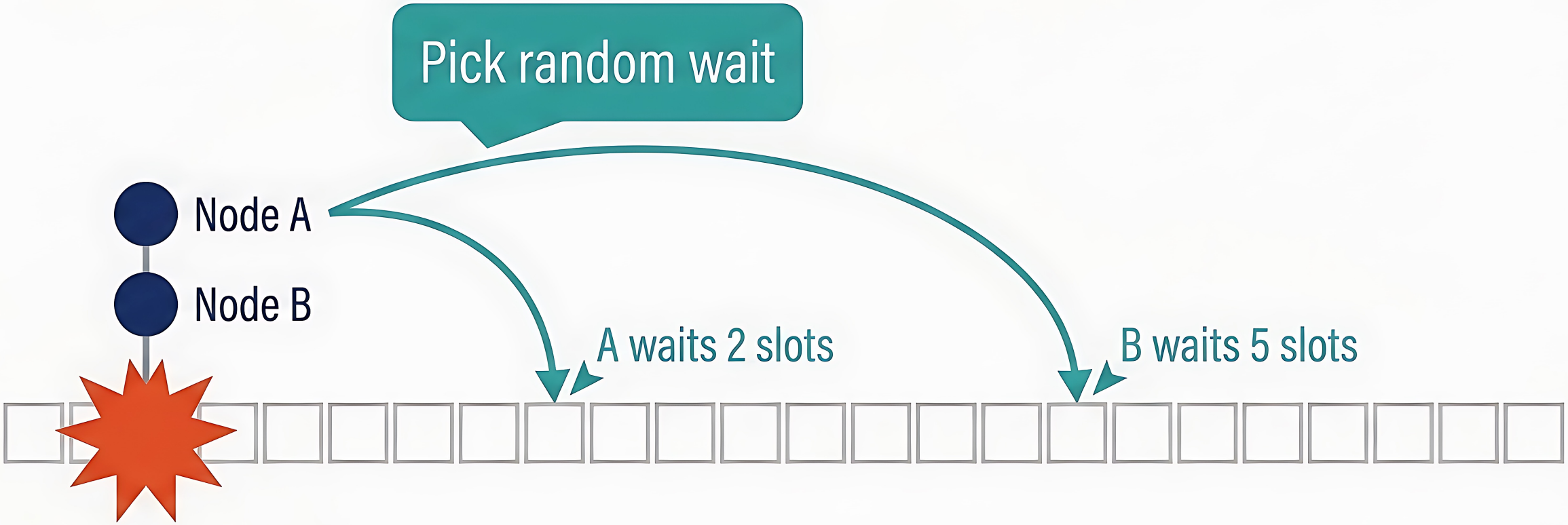
If the same two nodes collide and both retry immediately in the exact same way, they will just collide again.

Recovery needs spacing, not instant symmetry.



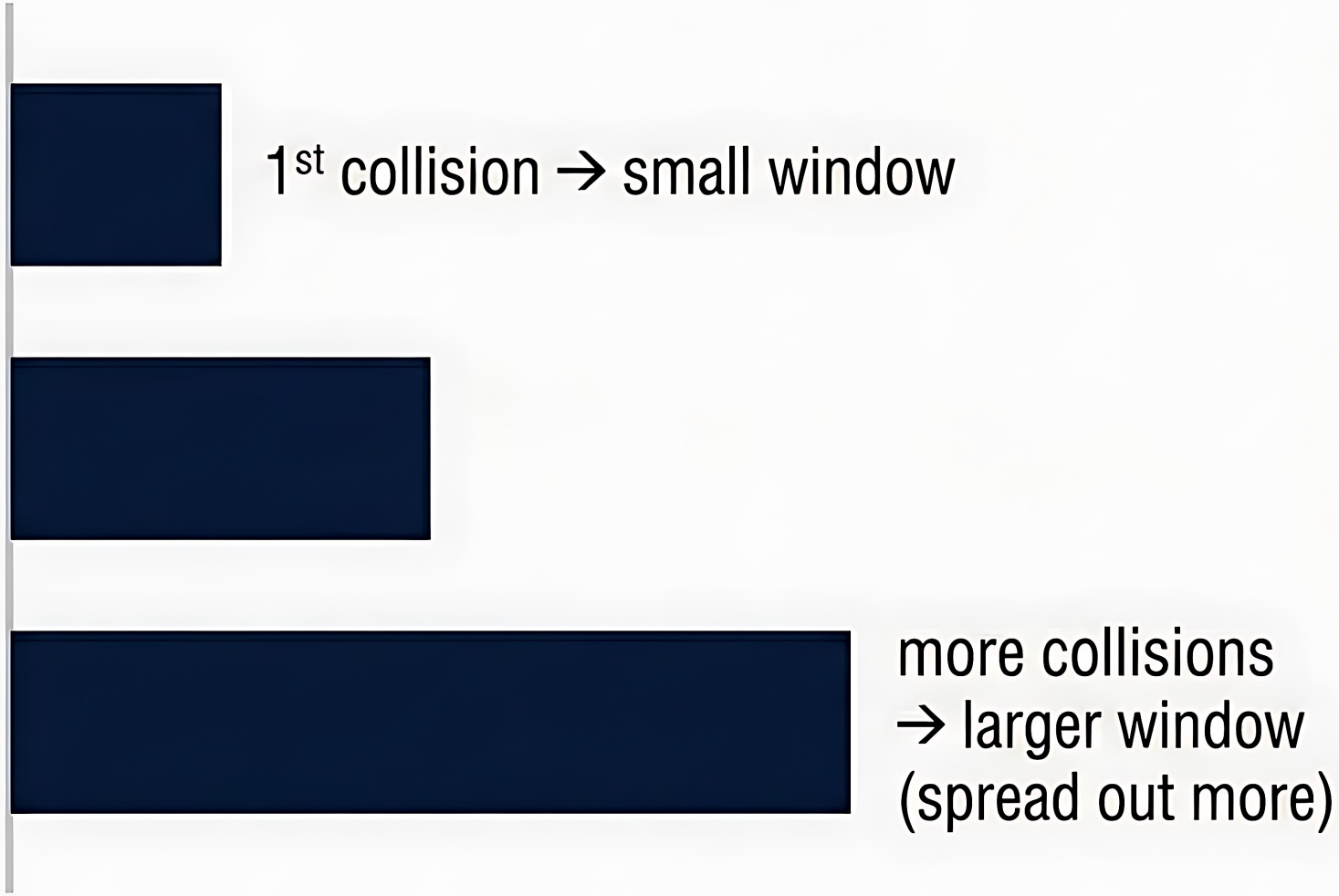
Random backoff breaks symmetry

Randomness helps competing nodes separate in time.



Binary exponential backoff scales to heavy contention

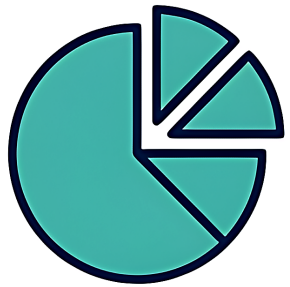
If collisions persist, contention is high. The protocol mathematically increases the wait range to reduce overlap probability.



The random-access story is a sequence of refinements

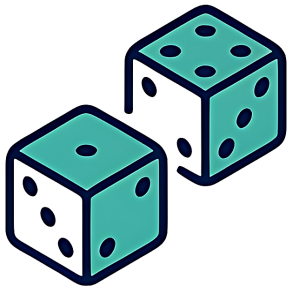
Every new protocol adds structure or information to fix a specific physical or systemic weakness of the previous one.

The Design Space: Three Approaches



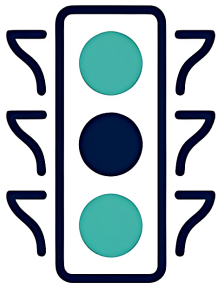
1. Partitioning

Split the resource ahead of time.



2. Random Access

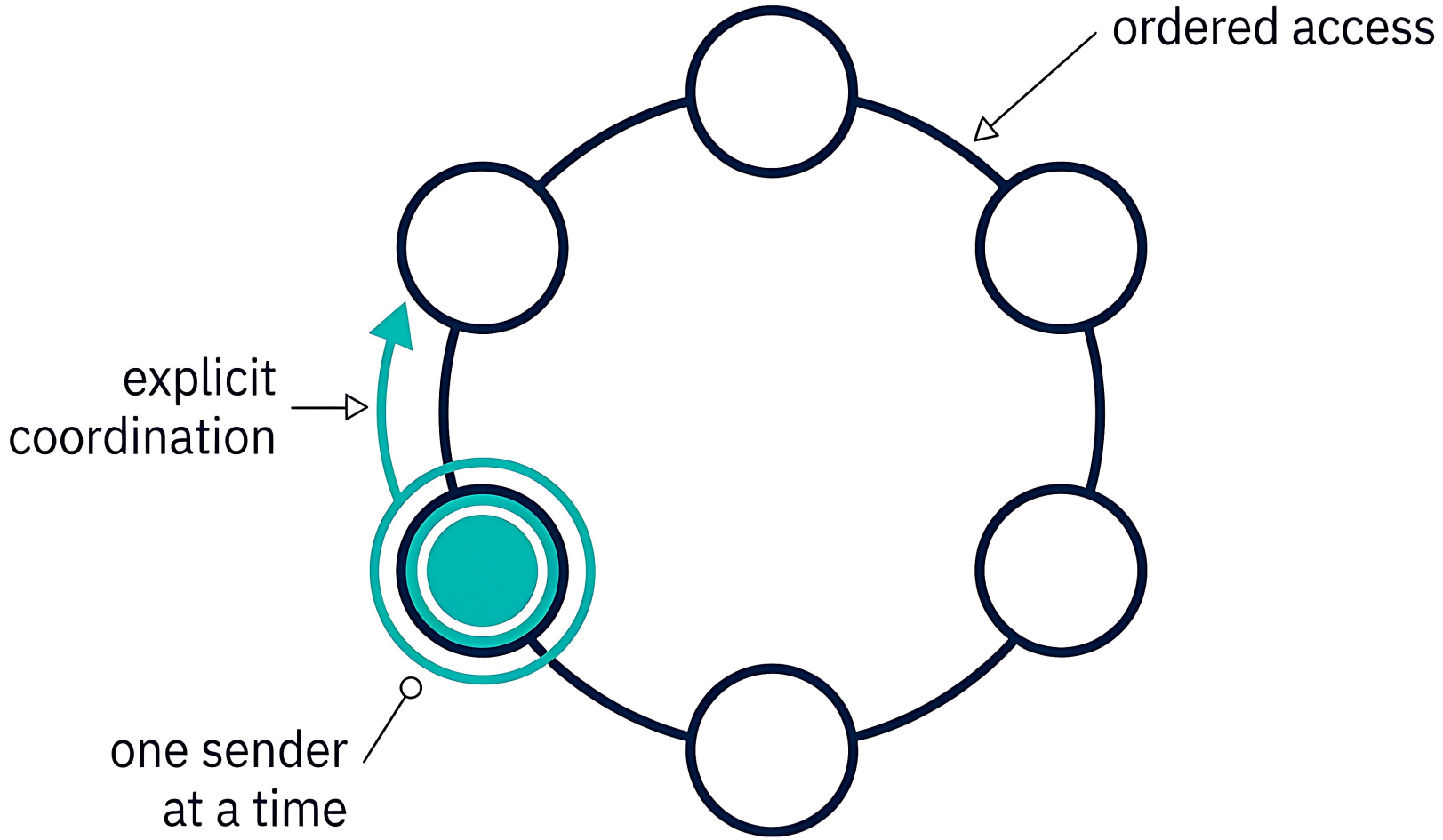
Compete only when needed.



3. Taking Turns

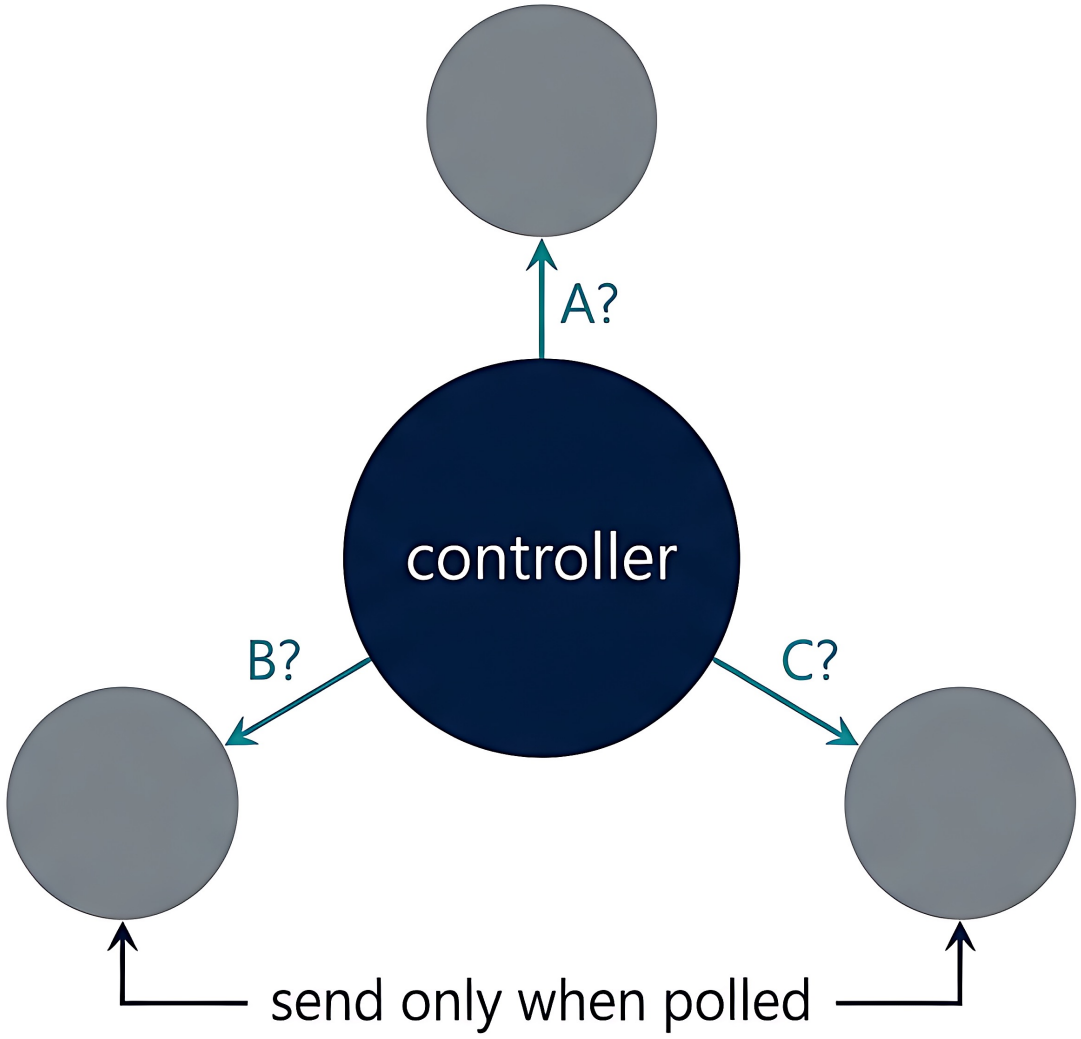
Explicitly decide whose turn it is.

A third option is to replace contention with explicit turns



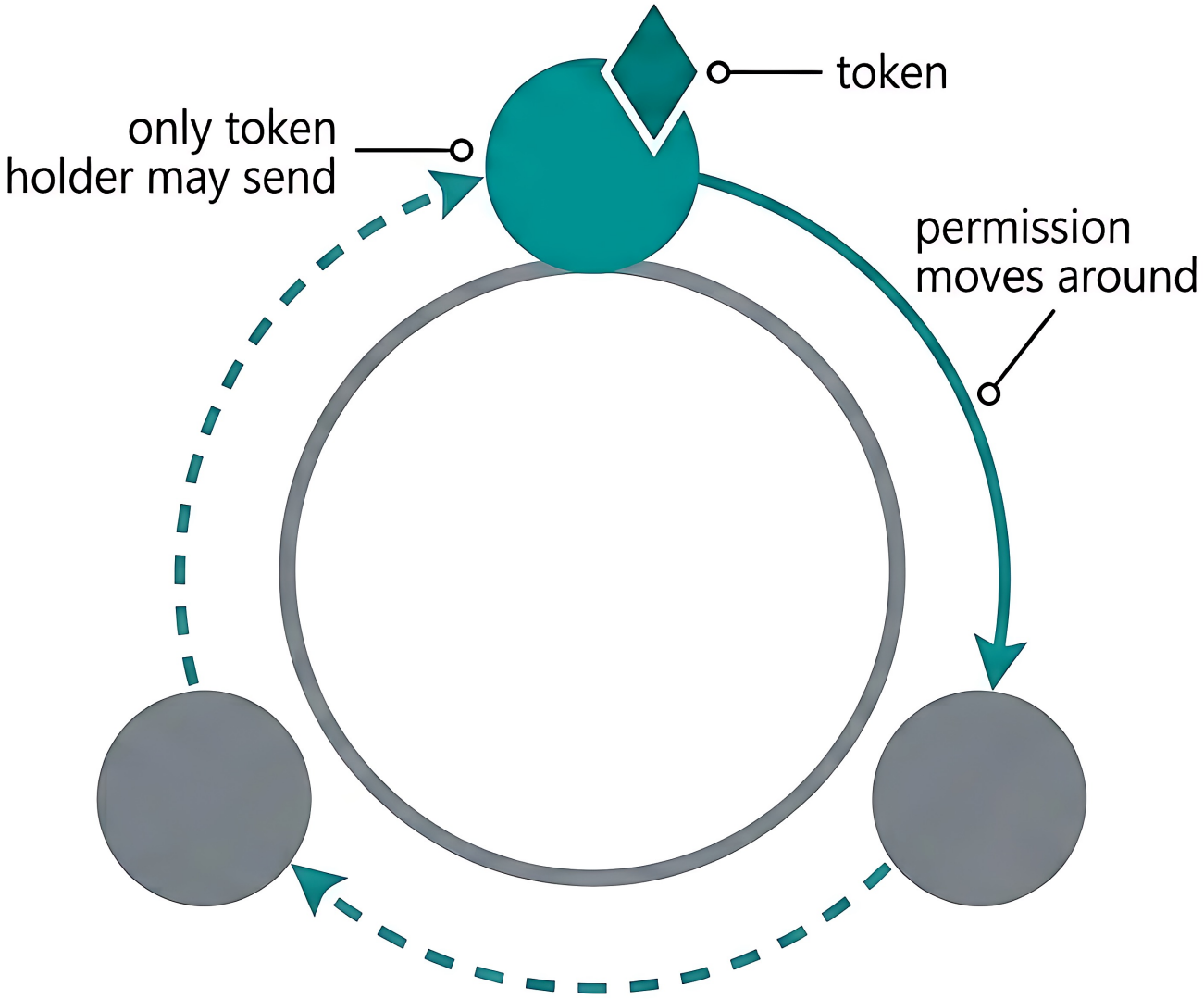
Polling uses a central controller

Polling avoids collisions through centralized coordination.



Token passing distributes the turn

Taking turns does not always require a central controller.



The MAC Comparison Matrix

	Partitioning	Random Access	Taking Turns
Collision Free	Yes	No	Yes
Bursty Traffic Efficiency	Poor	Excellent	Good
Fairness	High	Variable	Excellent
Coordination Overhead	Low	Low	High

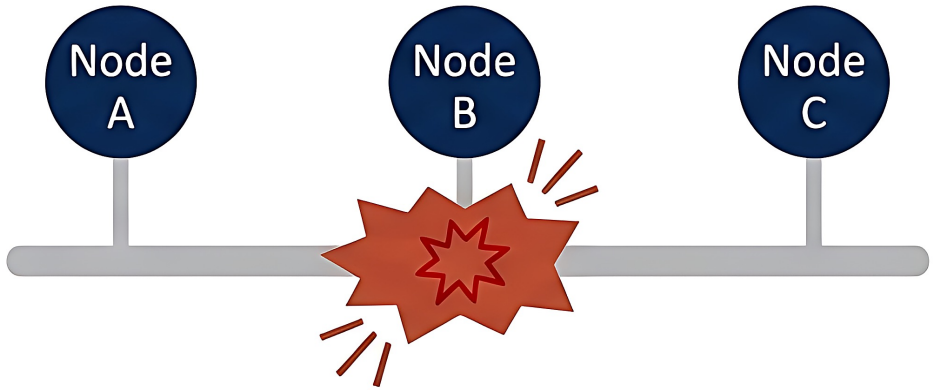
Real Systems: Redesigning the Architecture

Modern Ethernet removes the shared cable entirely.

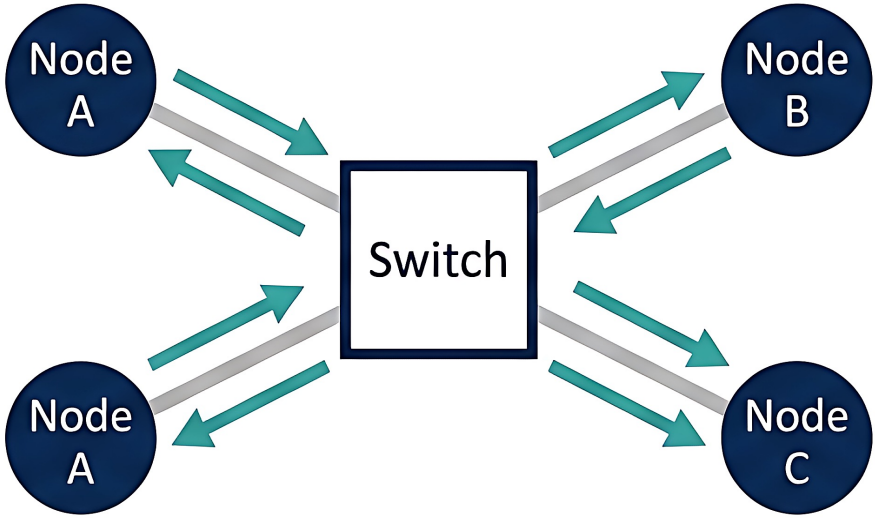
It uses switches and full-duplex, point-to-point links. The collision domain is physically eliminated.

System Lesson: Sometimes a protocol problem is solved by redesigning the system architecture.

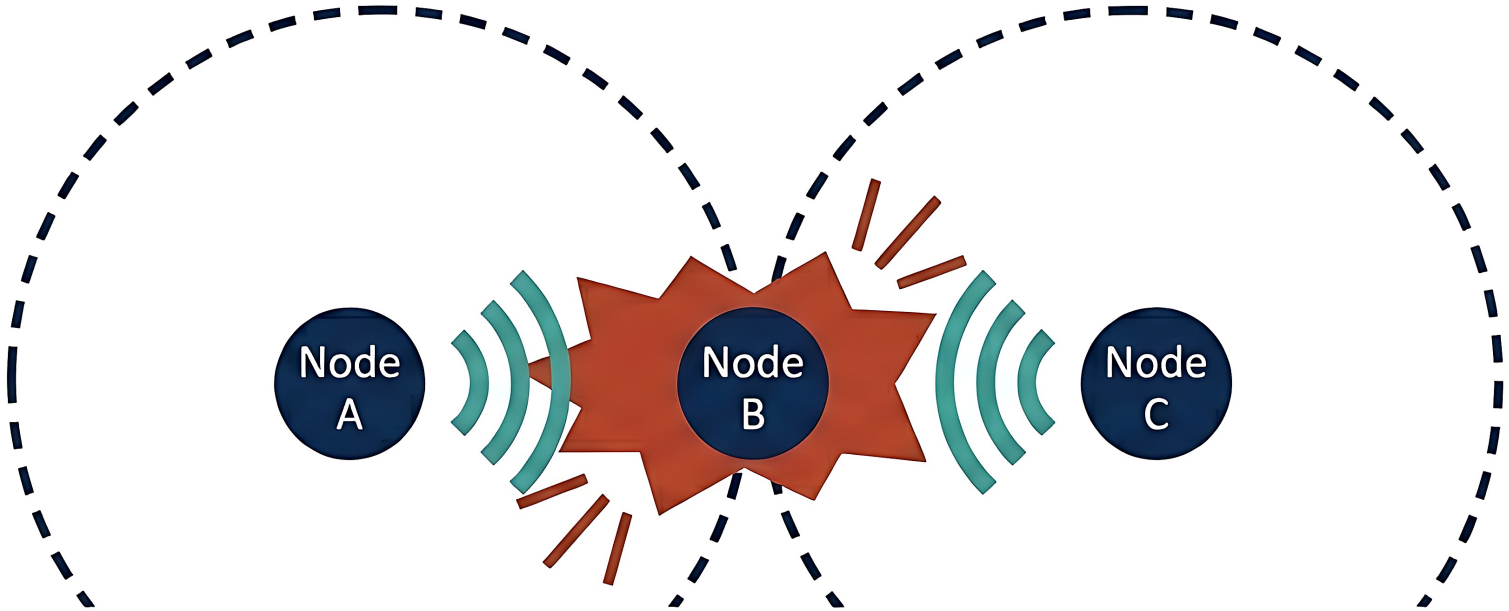
Before: Early Ethernet



After: Modern Switched Ethernet



Wireless & The Hidden Terminal Problem



Wireless nodes cannot easily listen while transmitting.

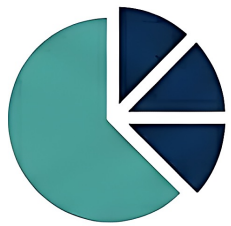
Hidden Terminal: A and C cannot hear each other, so both think the channel is idle. Their signals collide at B.

Local sensing is not enough to guarantee global safety.

Therefore, Wi-Fi emphasizes CSMA/CA (Collision Avoidance).

The Art of Multiple Access Control

"Multiple access control is the art of letting many independent senders share one communication medium without letting competition destroy efficiency."



Partitioning • Random Access • Taking Turns