A Fast Confirmation Rule for Ethereum

Aditya Asgaonkar
Offchain Labs

Francesco D'Amato
Ethereum Foundation

Roberto Saltini
Independent

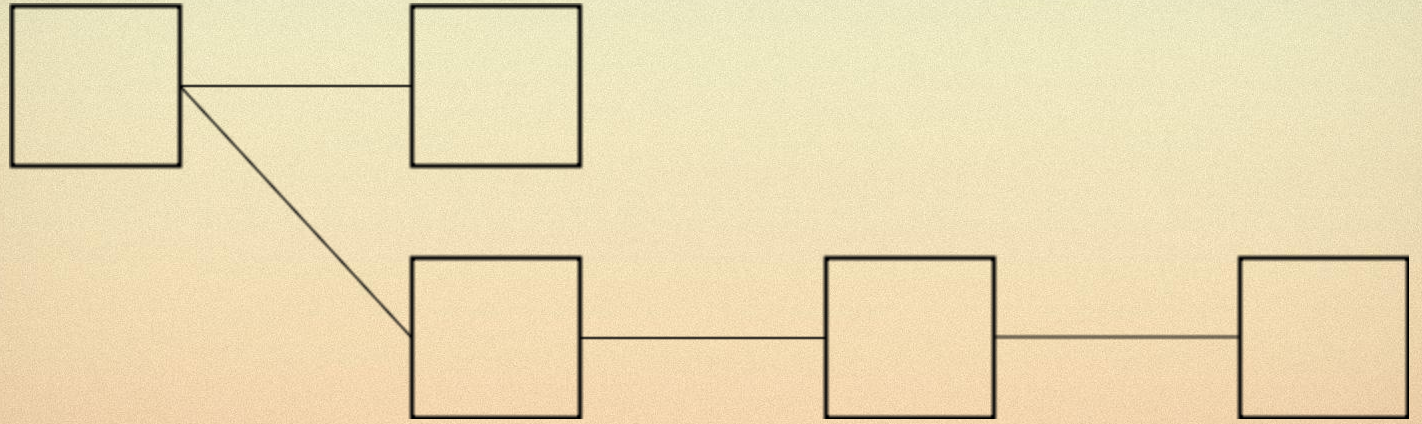
Luca Zanolini
Ethereum Foundation

Chenyi Zhang
University of Canterbury

Section 1

What is a Confirmation Rule?

Canonical Chain



Our Fast Confirmation Rule is an algorithm that enables nodes to identify blocks that will never leave the canonical chain under good network conditions

- It outputs whether a block is **confirmed**.

- Properties of the Confirmation Rule
  - **Safety**: If a block B is confirmed by a validator at a given time, it will be part of the canonical chain forever.
  - **Monotonicity**: Once a block B is confirmed at a given time, it will remain confirmed at all future times.

Our Fast Confirmation Rule is an algorithm that enables nodes to identify blocks that will never leave the canonical chain under good network conditions

- It outputs whether a block is **confirmed**.

- Properties of the Confirmation Rule

Our focus for today

  - **Safety**: If a block B is confirmed by a validator at a given time, it will be part of the canonical chain forever.
  - **Monotonicity**: Once a block B is confirmed at a given time, it will remain confirmed at all future times.

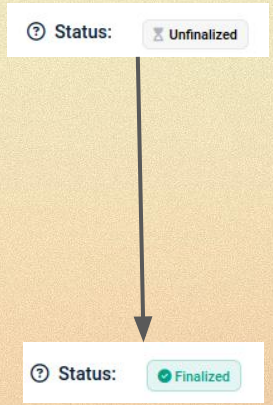
Section 2

Why do we need a Fast Confirmation Rule?

How do we know that a block will always be in the canonical chain?

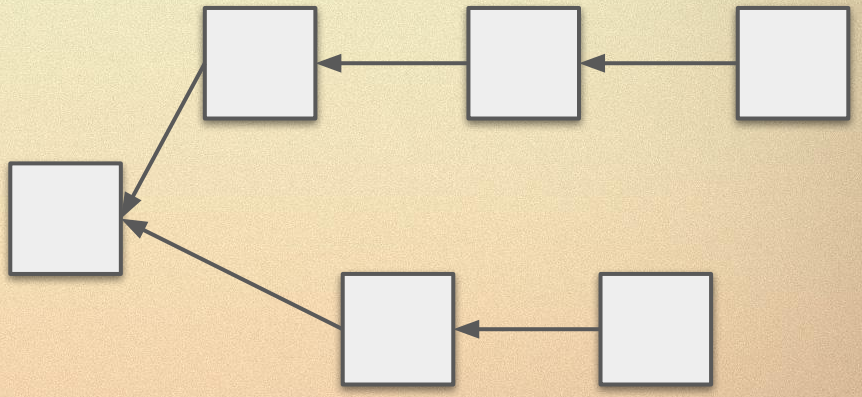
Today, we just have finalization

...which takes 13 min in the best-case scenario



What about block confirmations?

They were useful in PoW world, but not anymore



- For instance, because of balancing attacks

What does a Fast Confirmation Rule give us?

- **Improve Ethereum's UX:**
  - For small value transactions, we don't have to wait for finalization.
  - Reduces risk trade reversals for exchanges that allow to optimistically trade immediately upon deposit.
  - Improve wallets reliability: Some wallets tell users that a transaction is "confirmed" as soon as it is included in a block.

- **Useful for PBS:** Provides block builders with an indication of whether the block that they are building upon is unlikely to be reorged out.

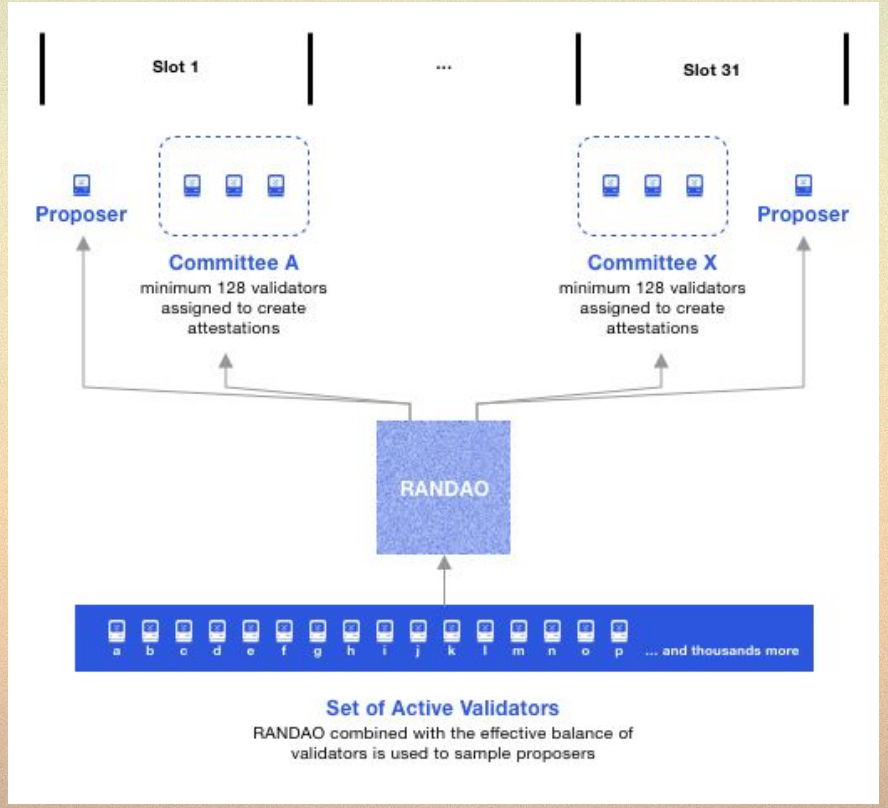
Section 3

Quick recap of the Ethereum's protocol

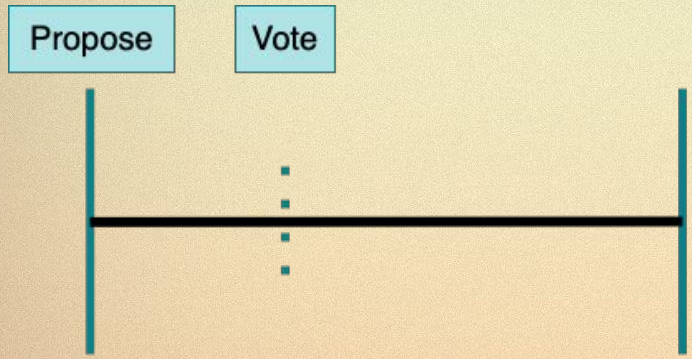
Gasper Consensus Protocol

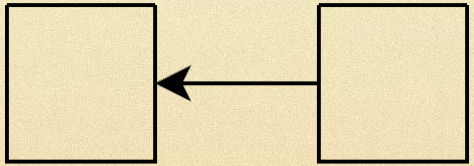
- Proof-of-Stake (PoS) consensus protocol.
- **LMD-GHOST**: A synchronous consensus protocol that determines the canonical chain.
- **Casper FFG**: A partially synchronous gadget that finalizes the blocks outputted by LMD-GHOST.

Gasper

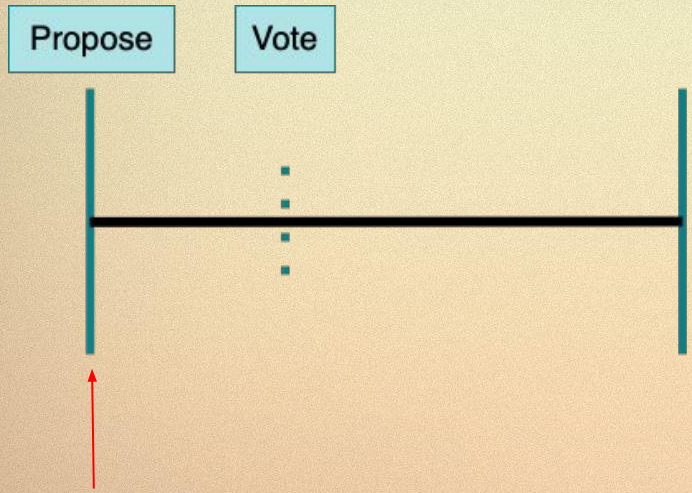

Slot 1
...
Slot 31
Proposer
Committee A
minimum 128 validators assigned to create attestations
Committee X
minimum 128 validators assigned to create attestations
Proposer
RANDAO
a b c d e f g h i j k l m n o p
... and thousands more
Set of Active Validators
RANDAO combined with the effective balance of validators is used to sample proposers

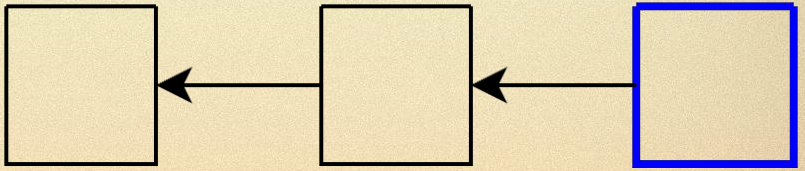
LMD-GHOST Protocol



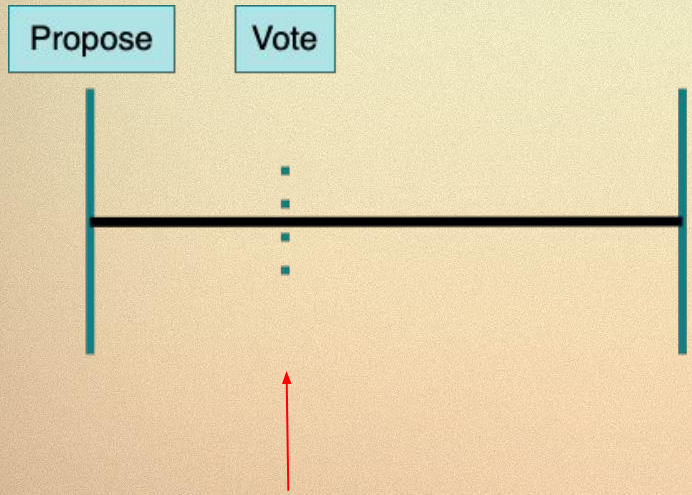


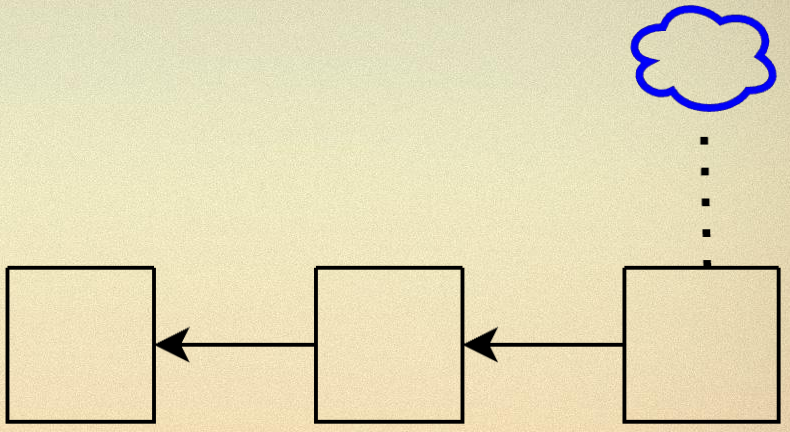
LMD-GHOST Protocol



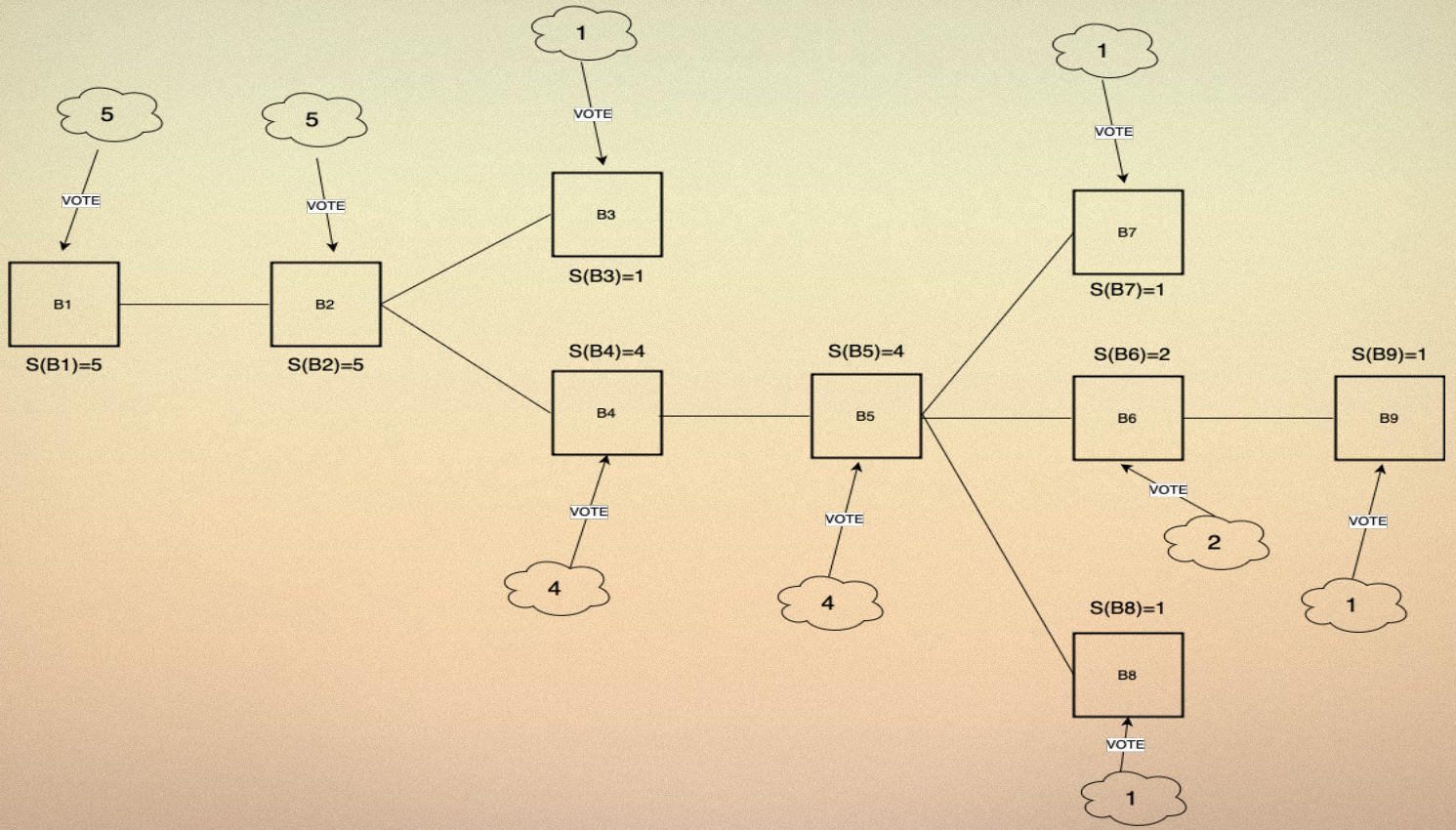


LMD-GHOST Protocol

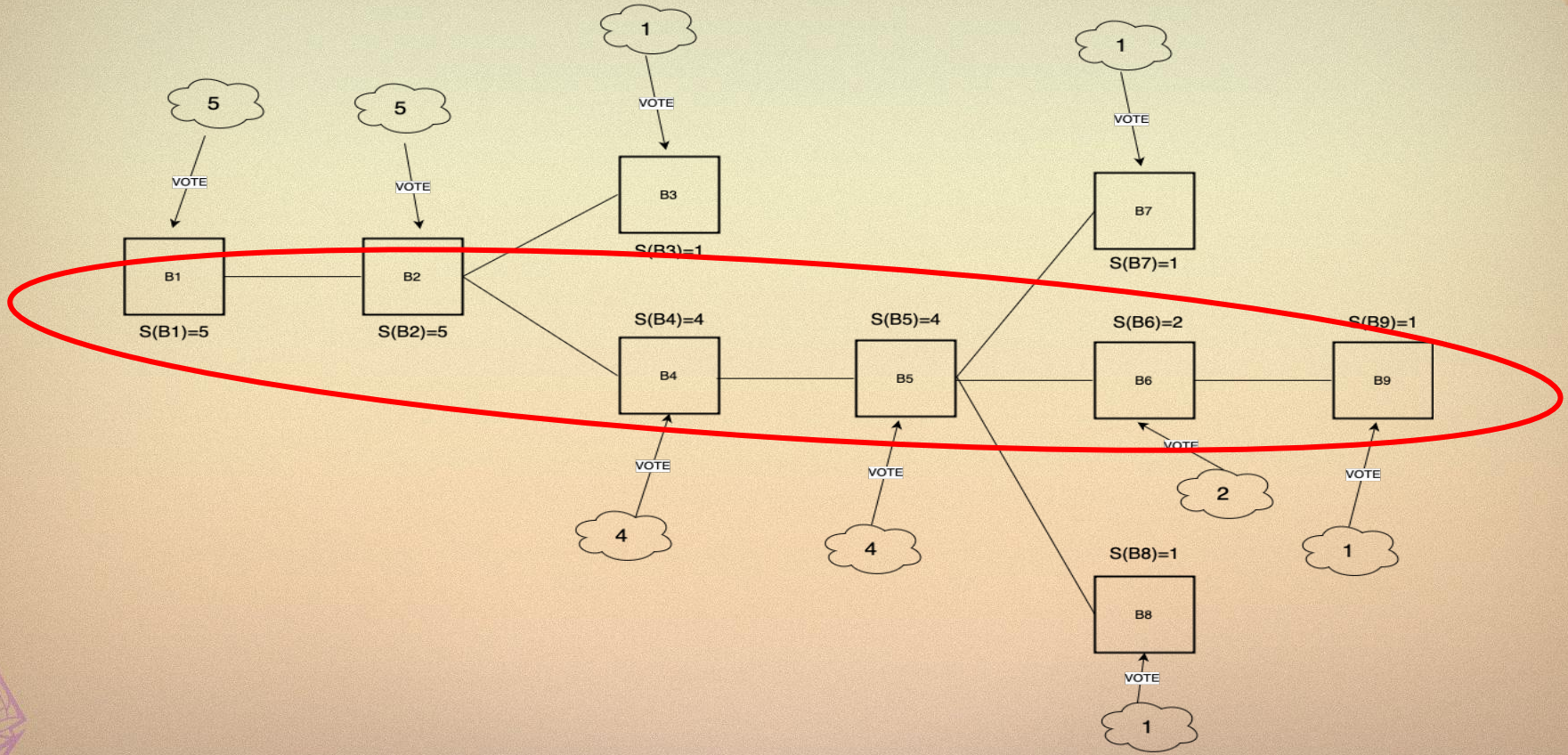




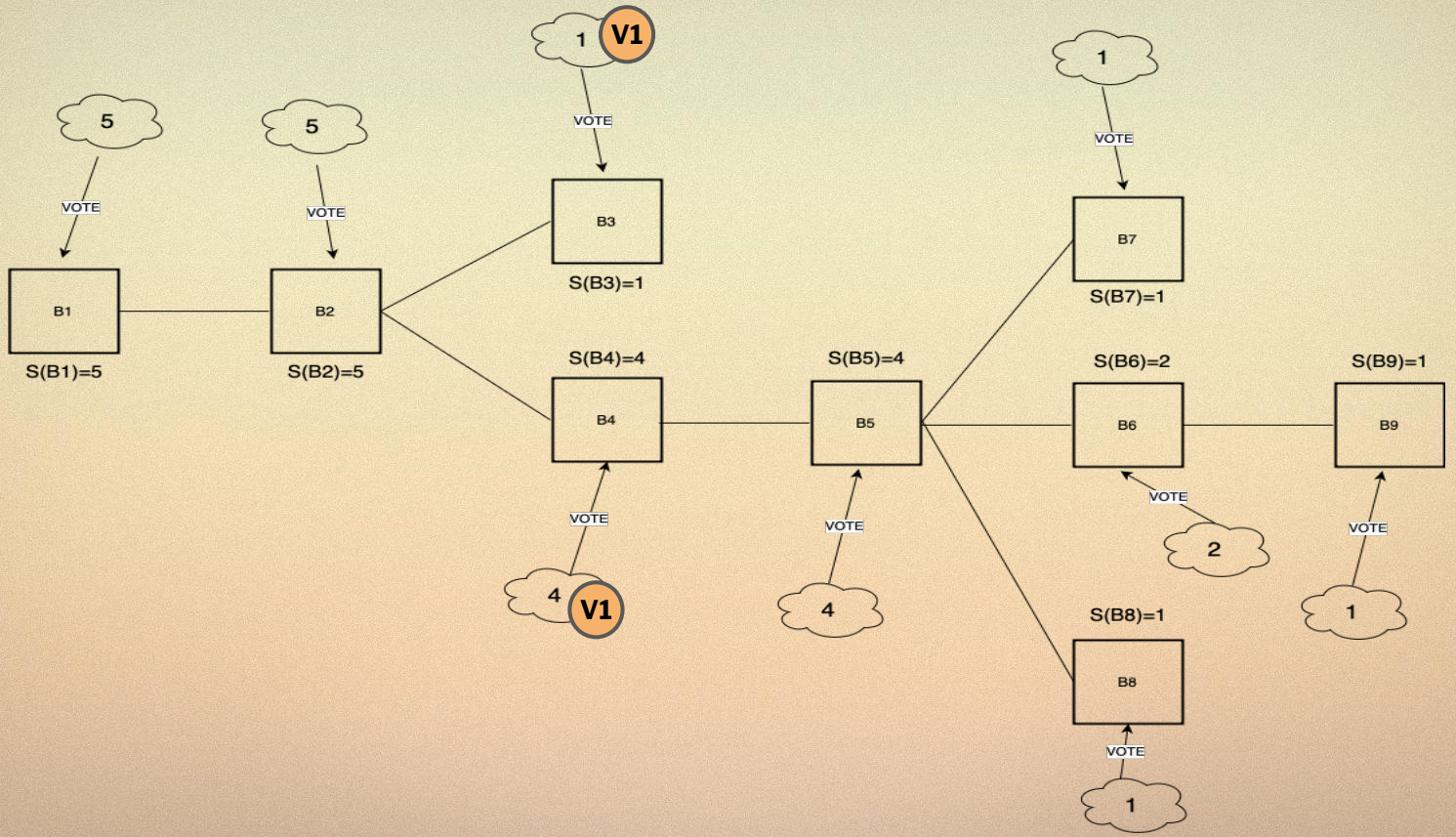
LMD-GHOST Fork-Choice function



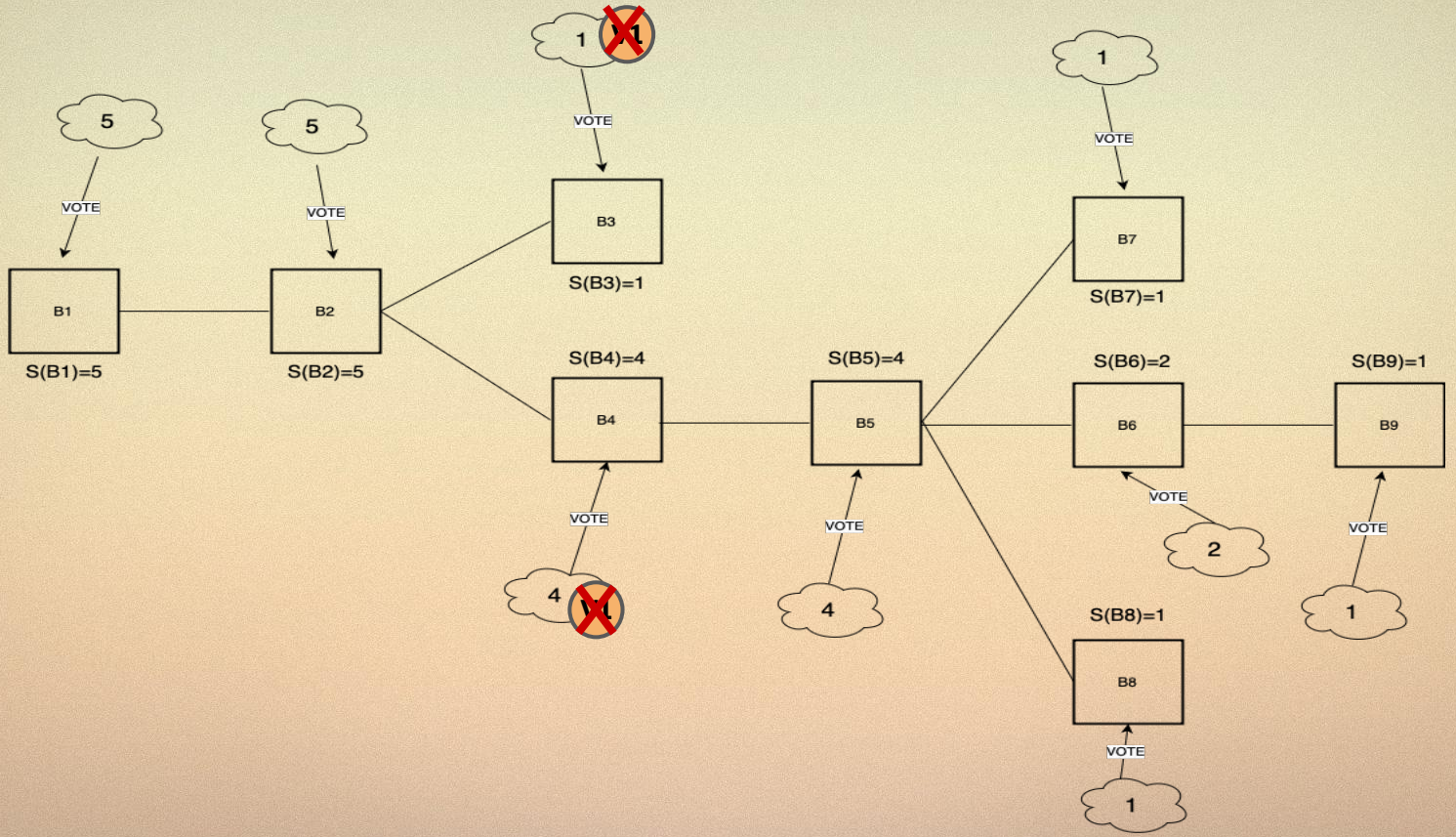
LMD-GHOST Fork-Choice function



LMD-GHOST Fork-Choice function



LMD-GHOST Fork-Choice function

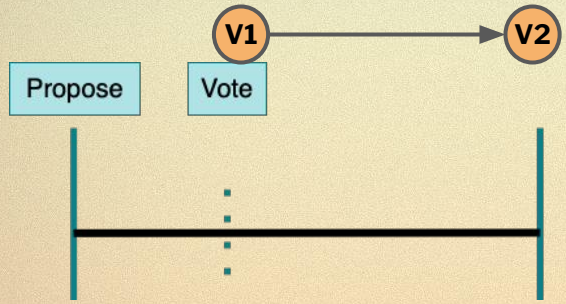


Section 4

A Confirmation Rule Algorithm for LMD-GHOST

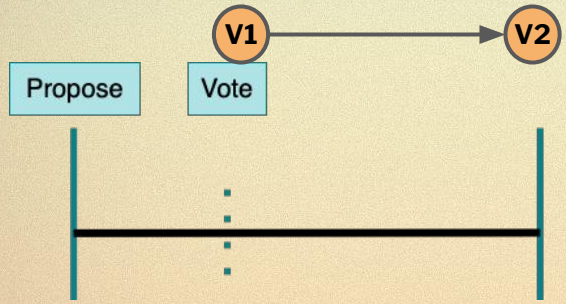
Assumptions

- LMD-GHOST votes are delivered by the end of a slot

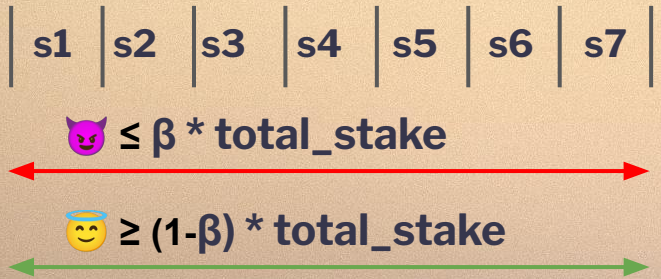


Assumptions

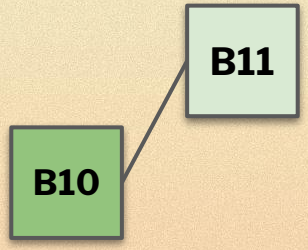
- LMD-GHOST votes are delivered by the end of a slot



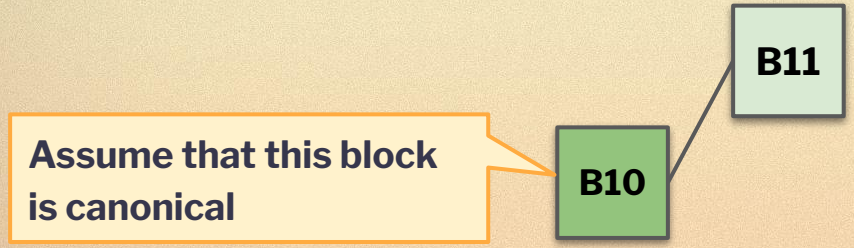
- Max fraction β of the stake of any set of committees is dishonest



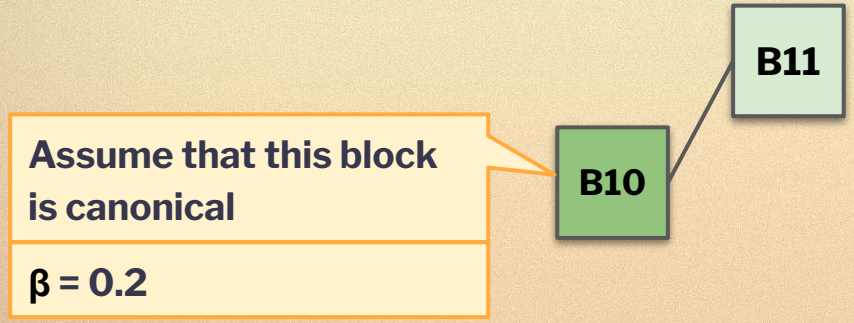
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



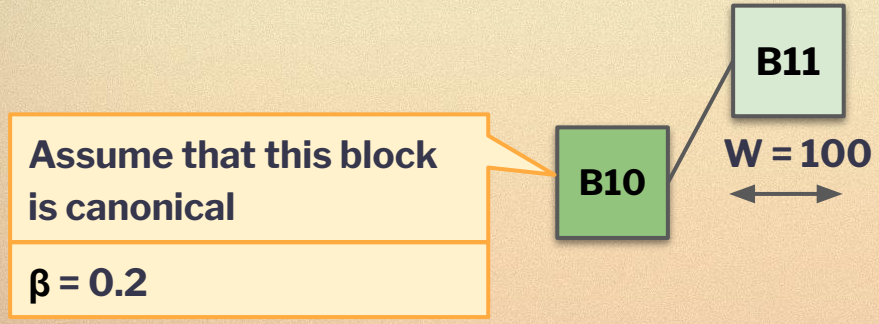
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



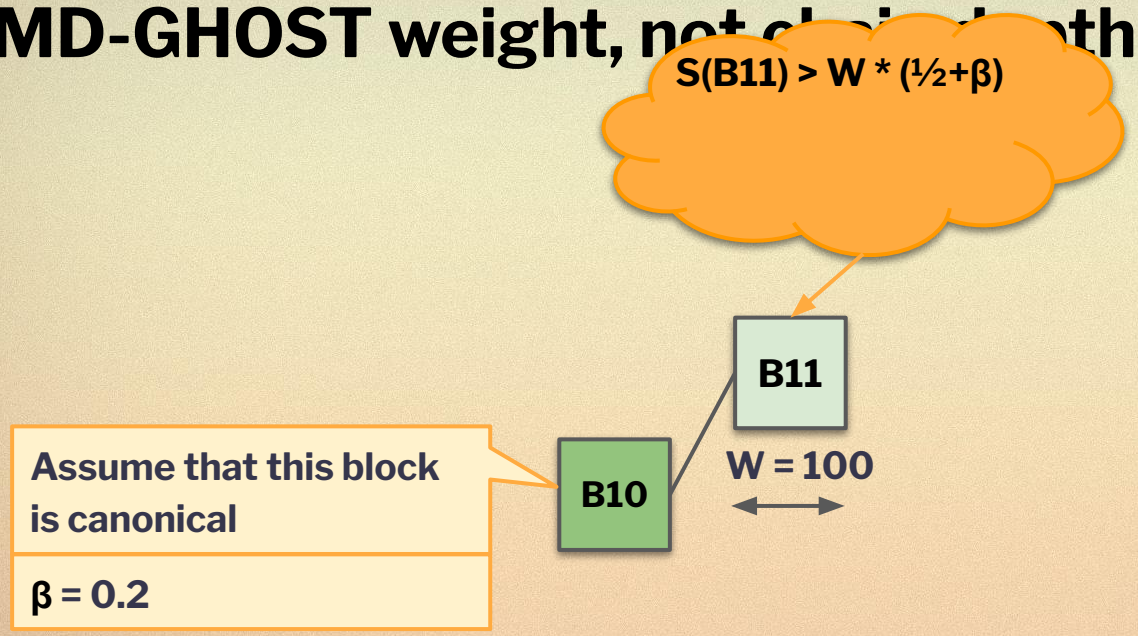
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



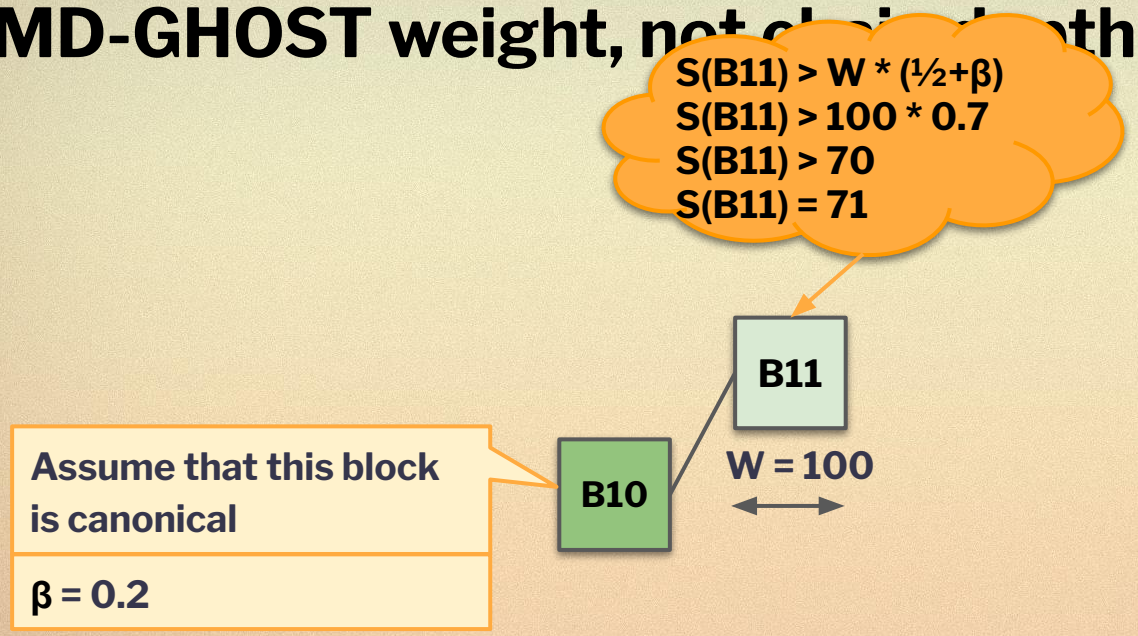
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



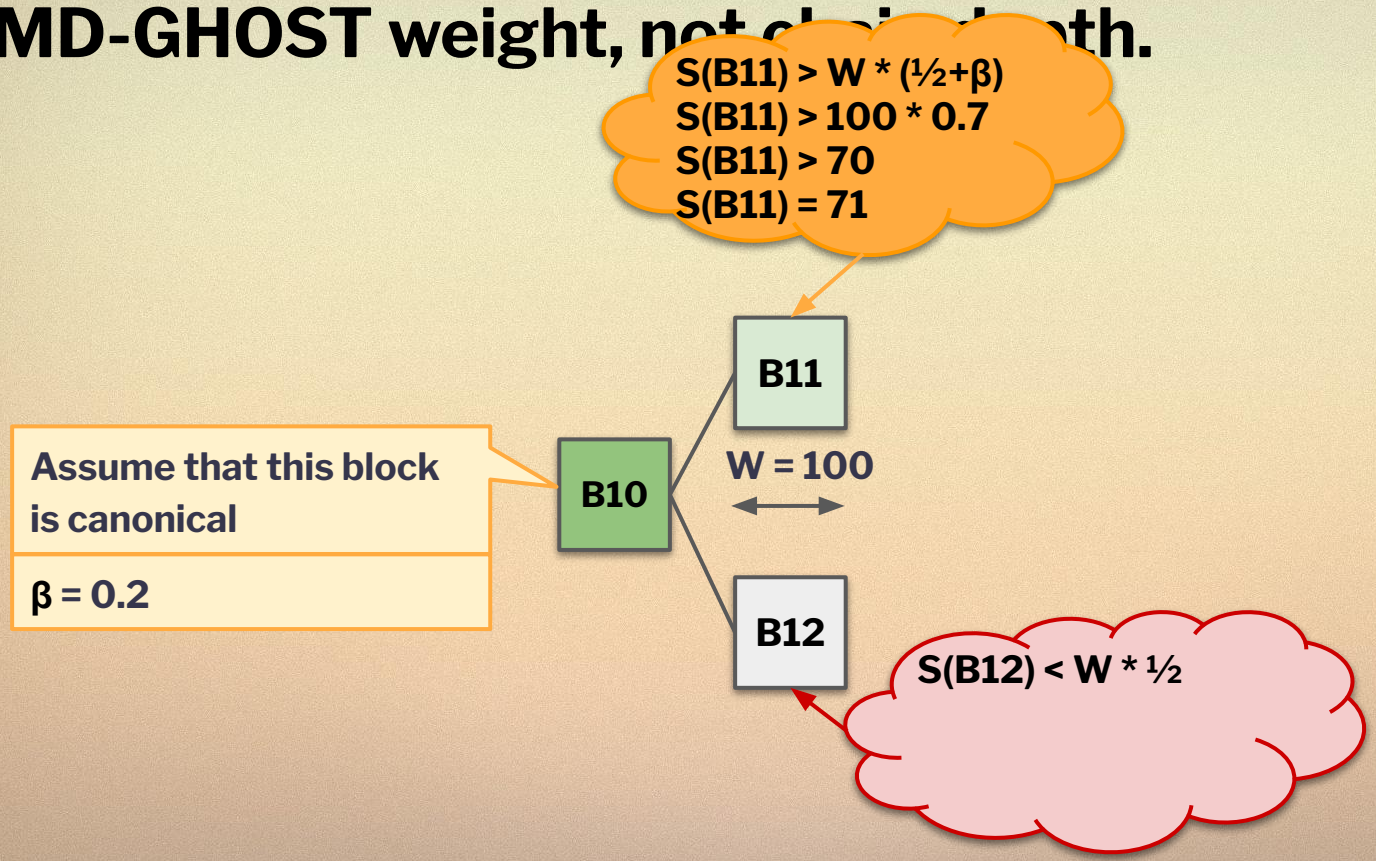
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



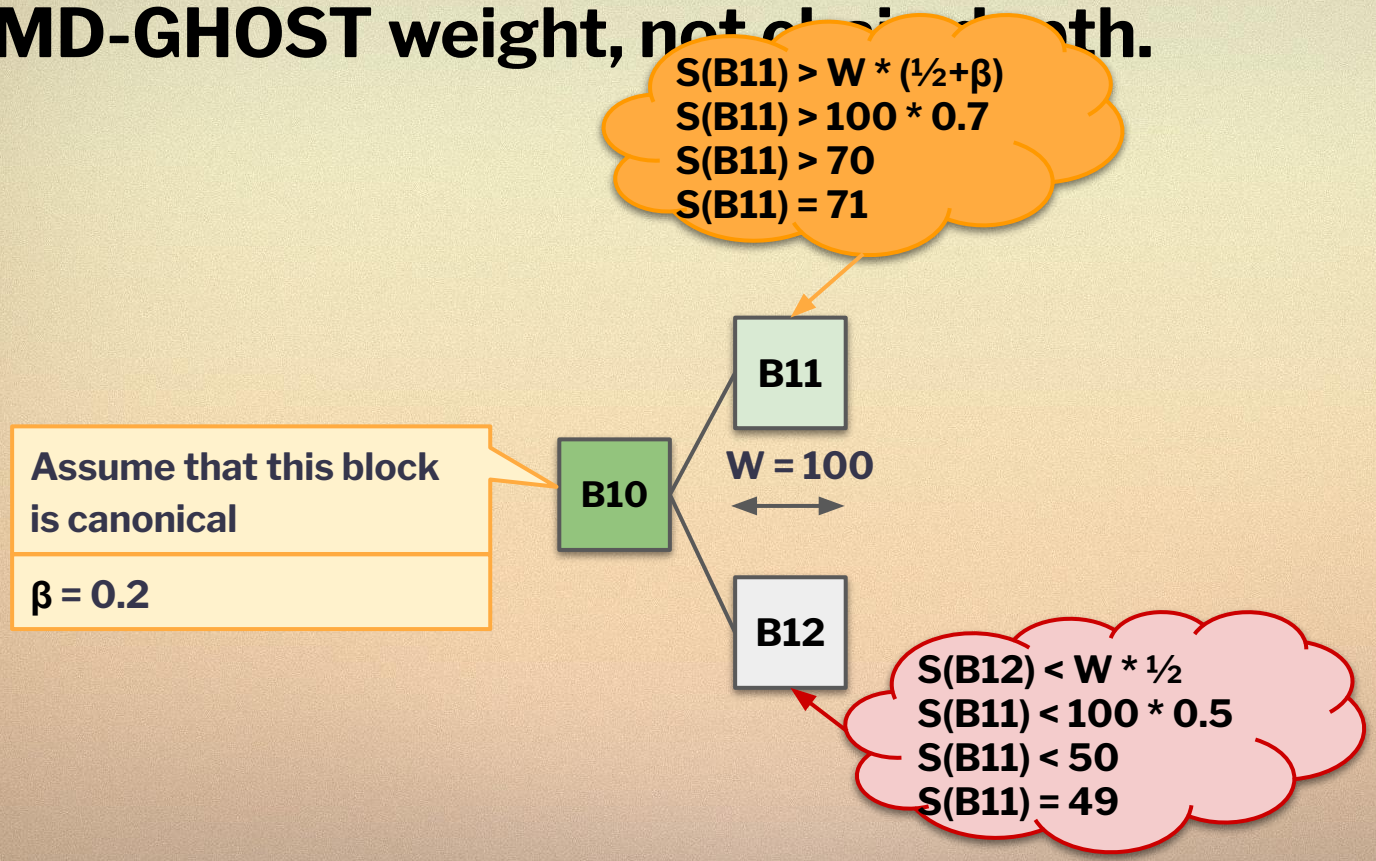
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain length.



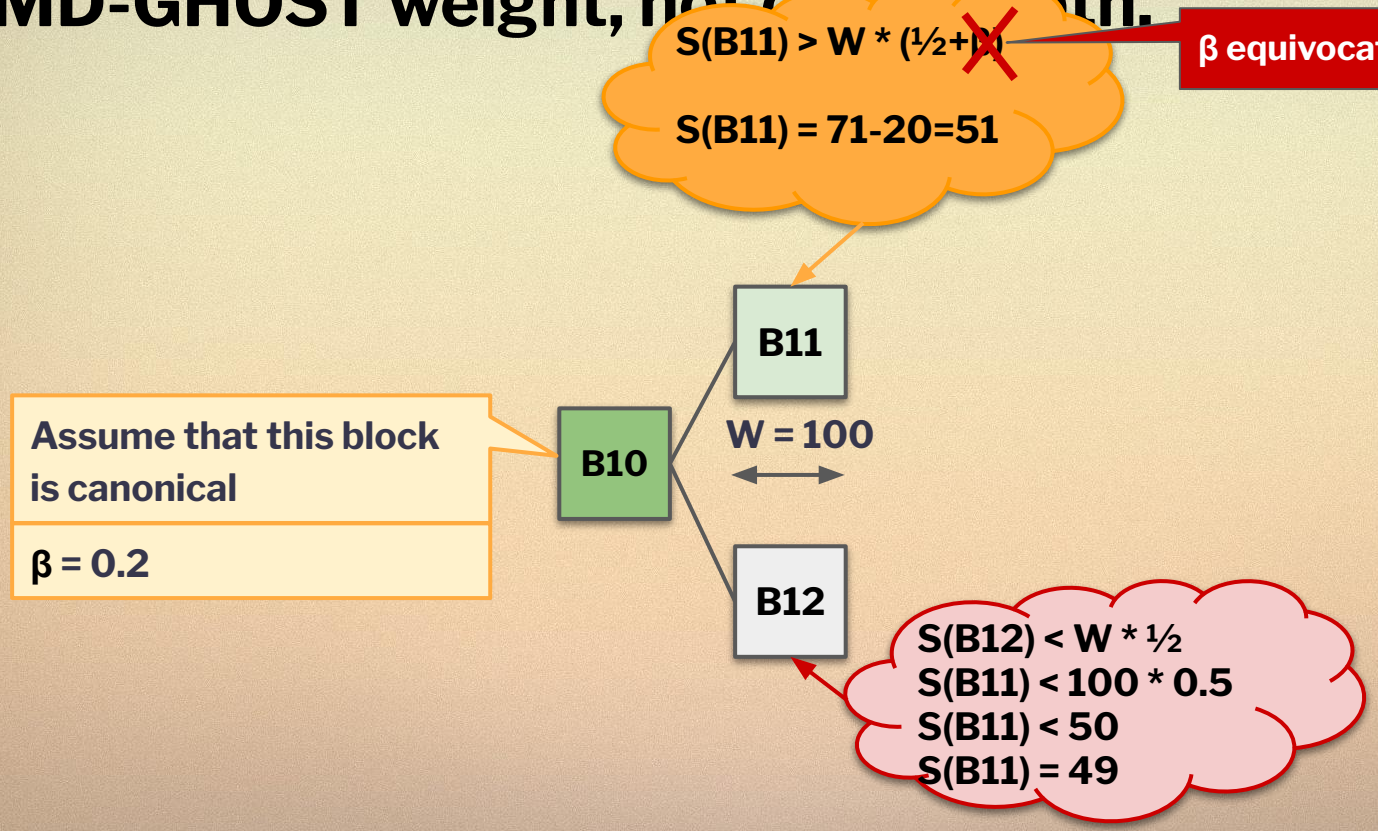
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



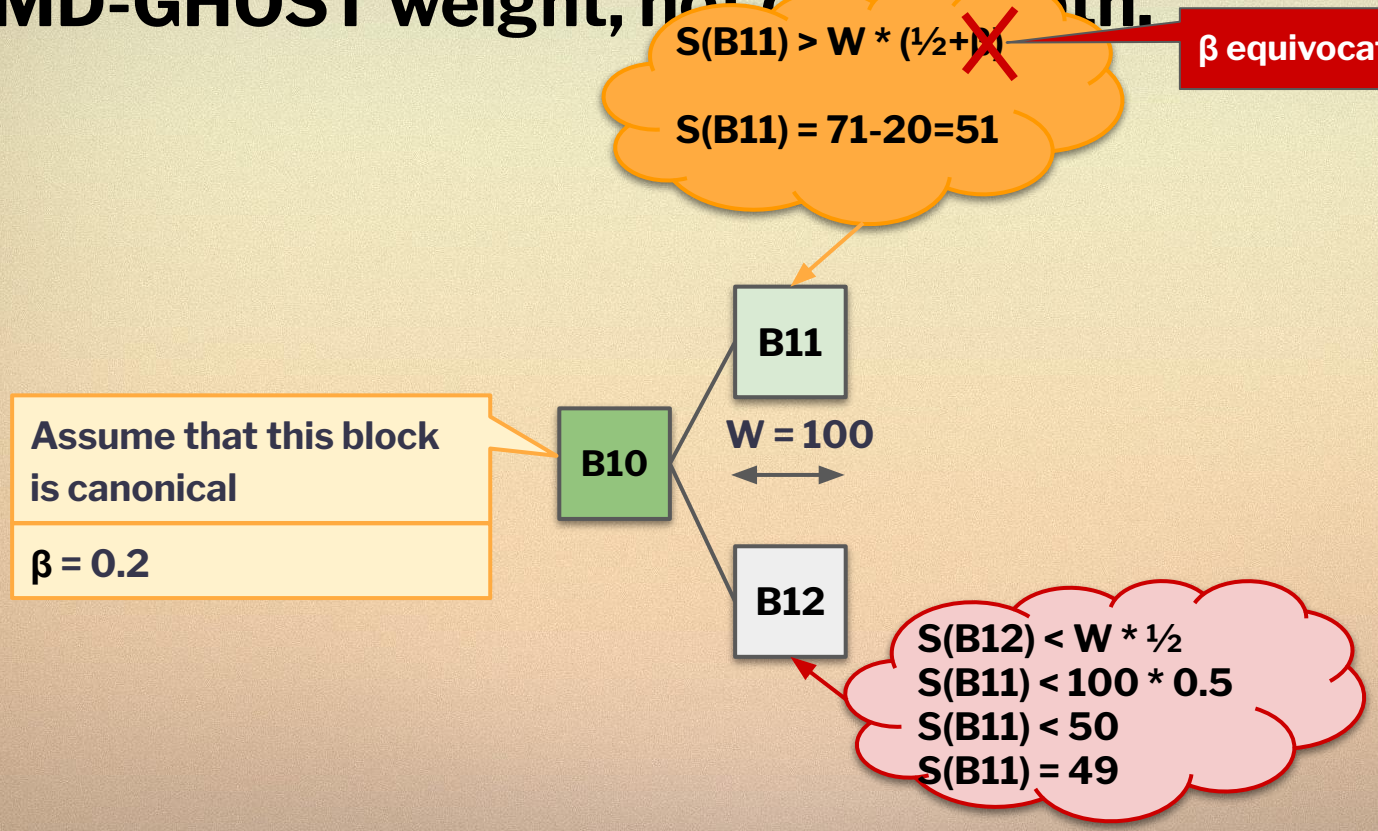
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



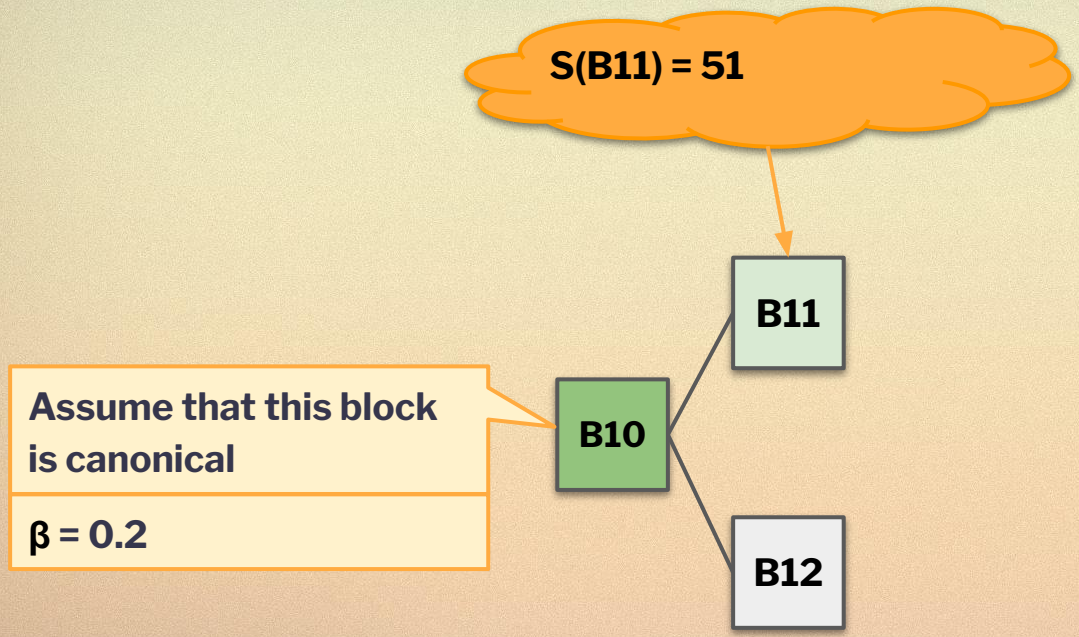
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



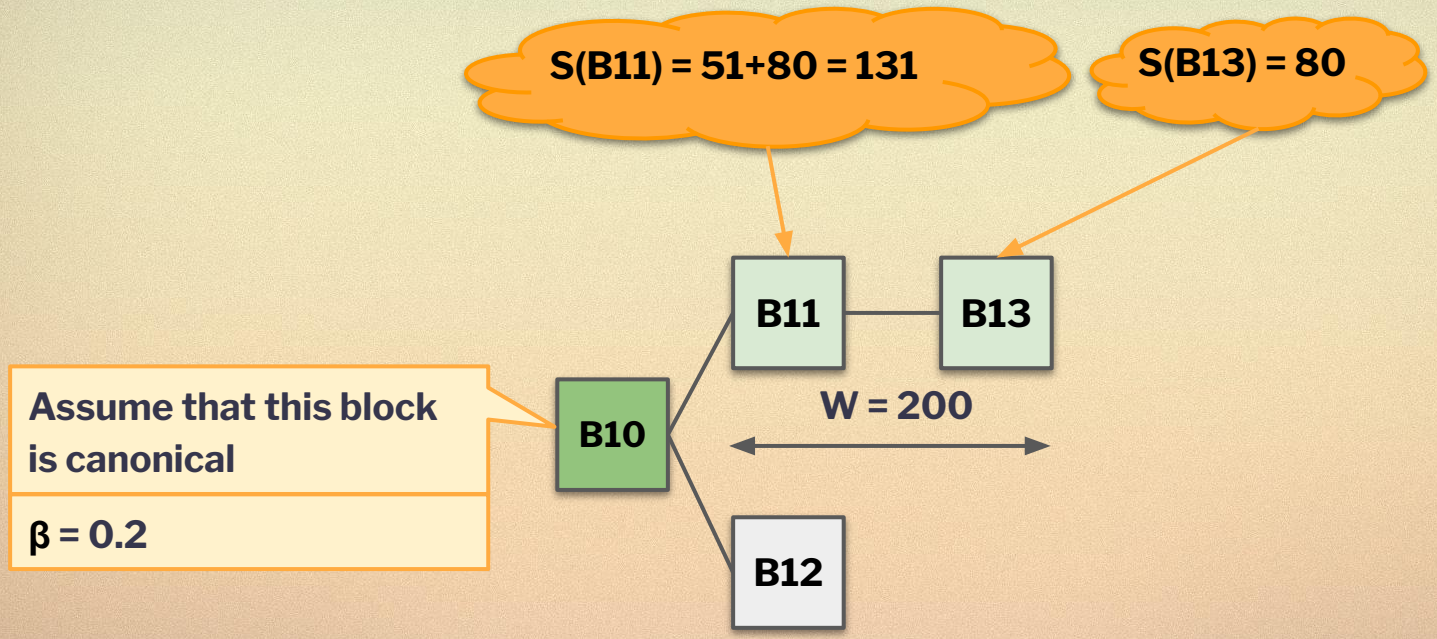
A Confirmation Rule for Ethereum is about LMD-GHOST weight, not chain depth.



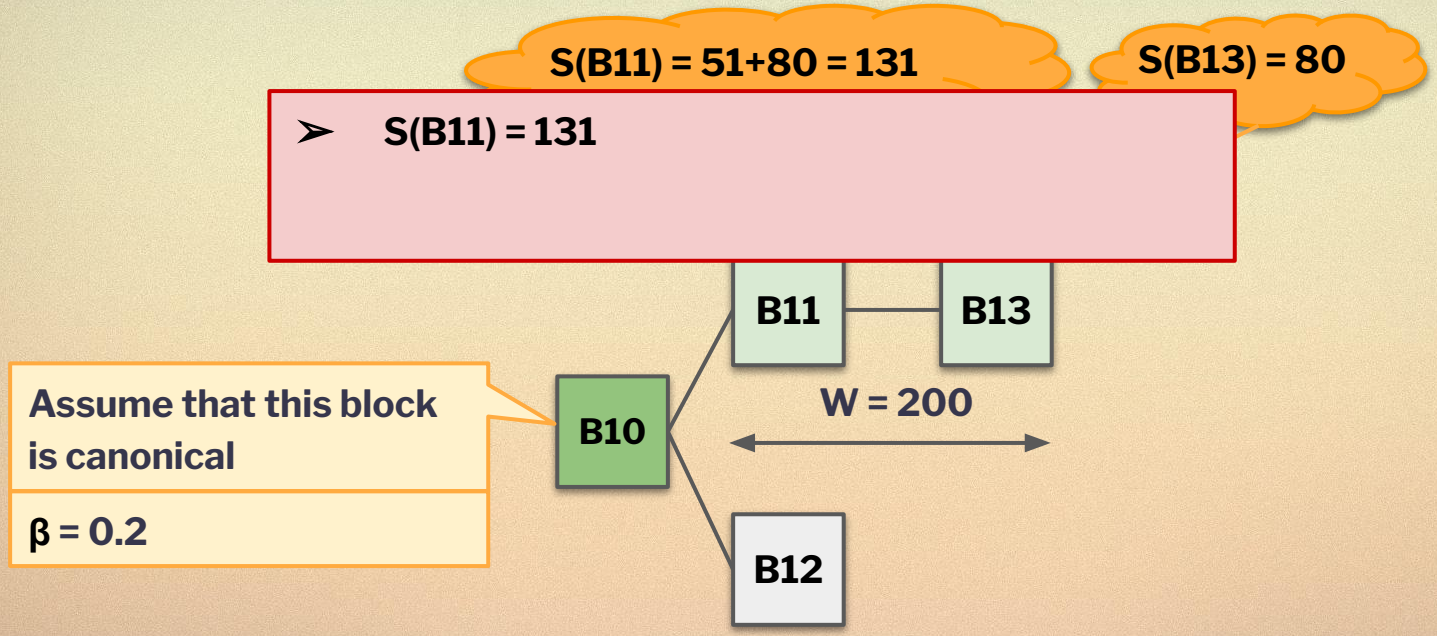
What about when we move to the next slot?



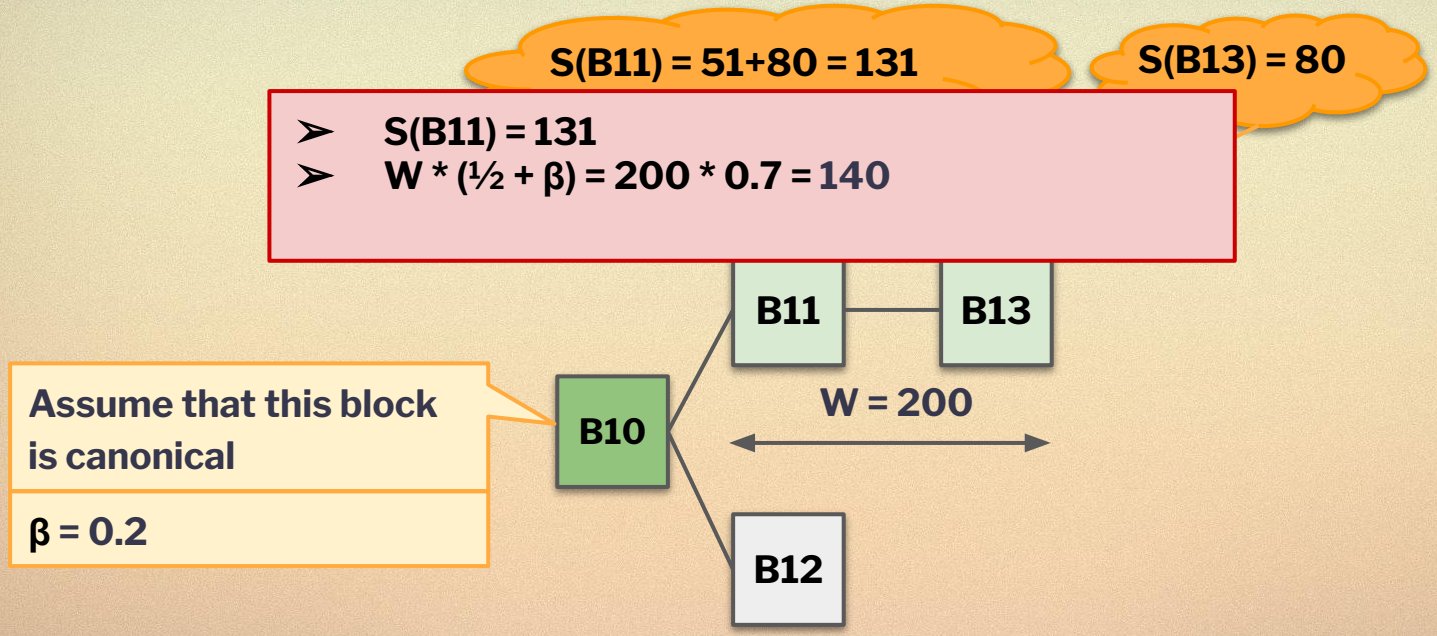
What about when we move to the next slot?



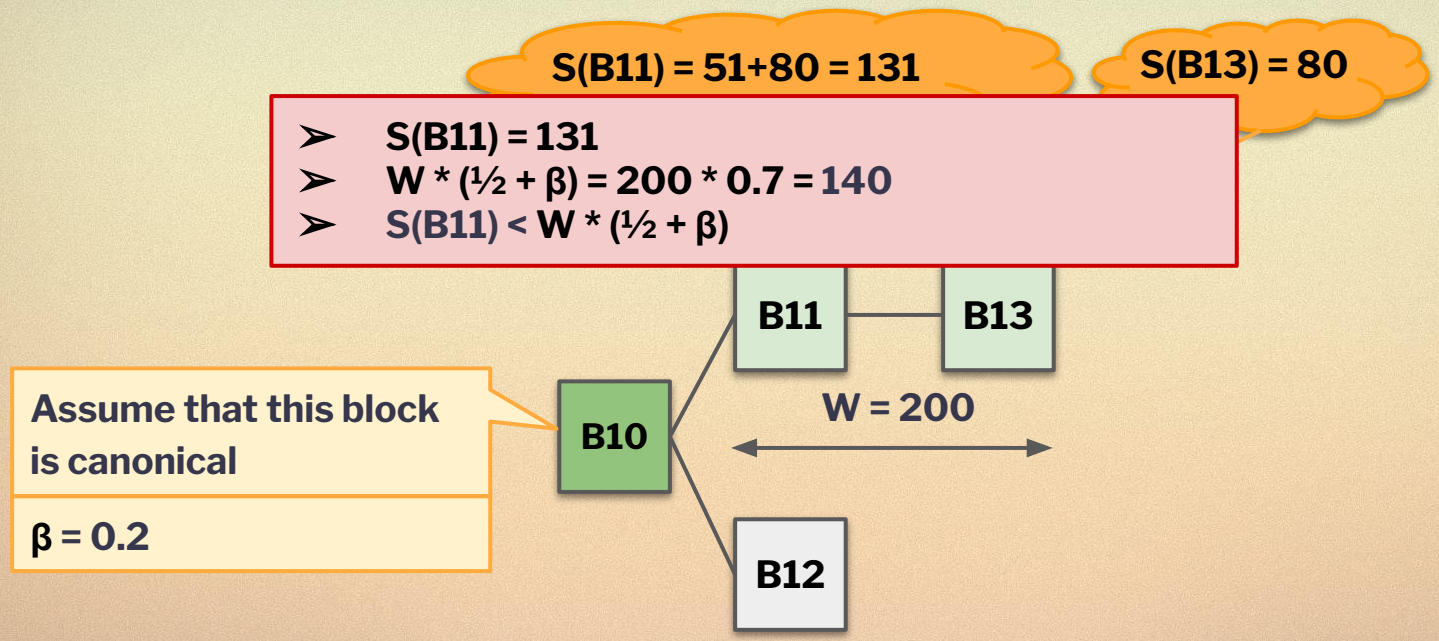
What about when we move to the next slot?



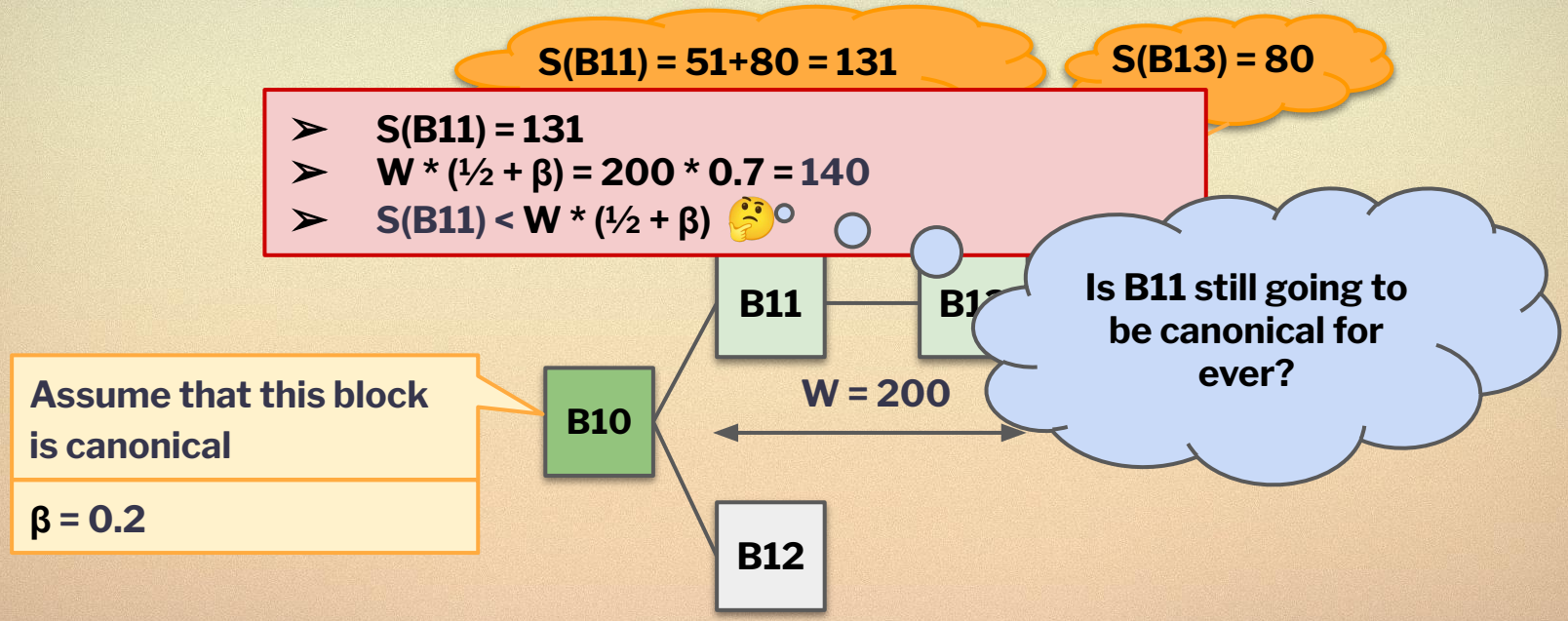
What about when we move to the next slot?



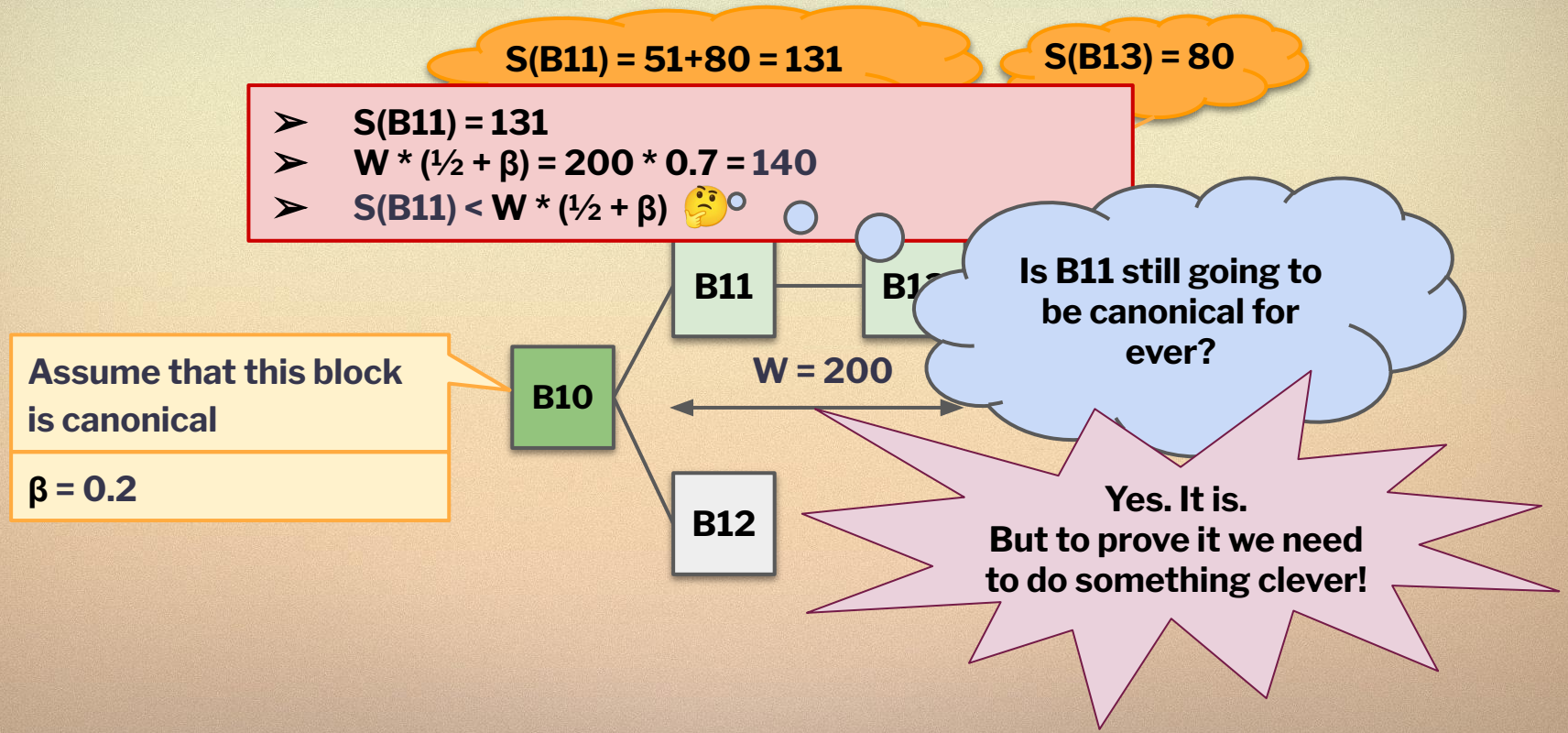
What about when we move to the next slot?



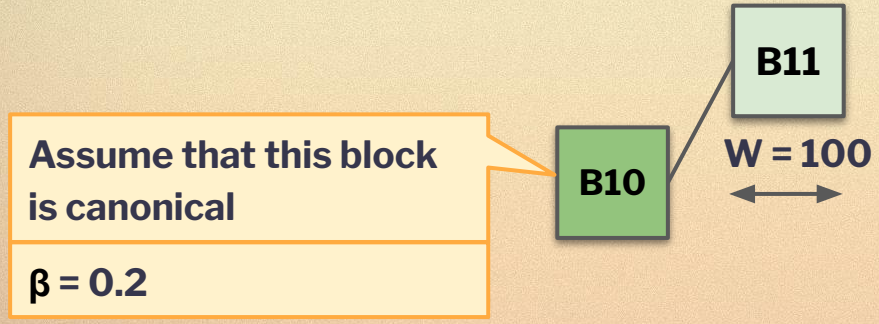
What about when we move to the next slot?



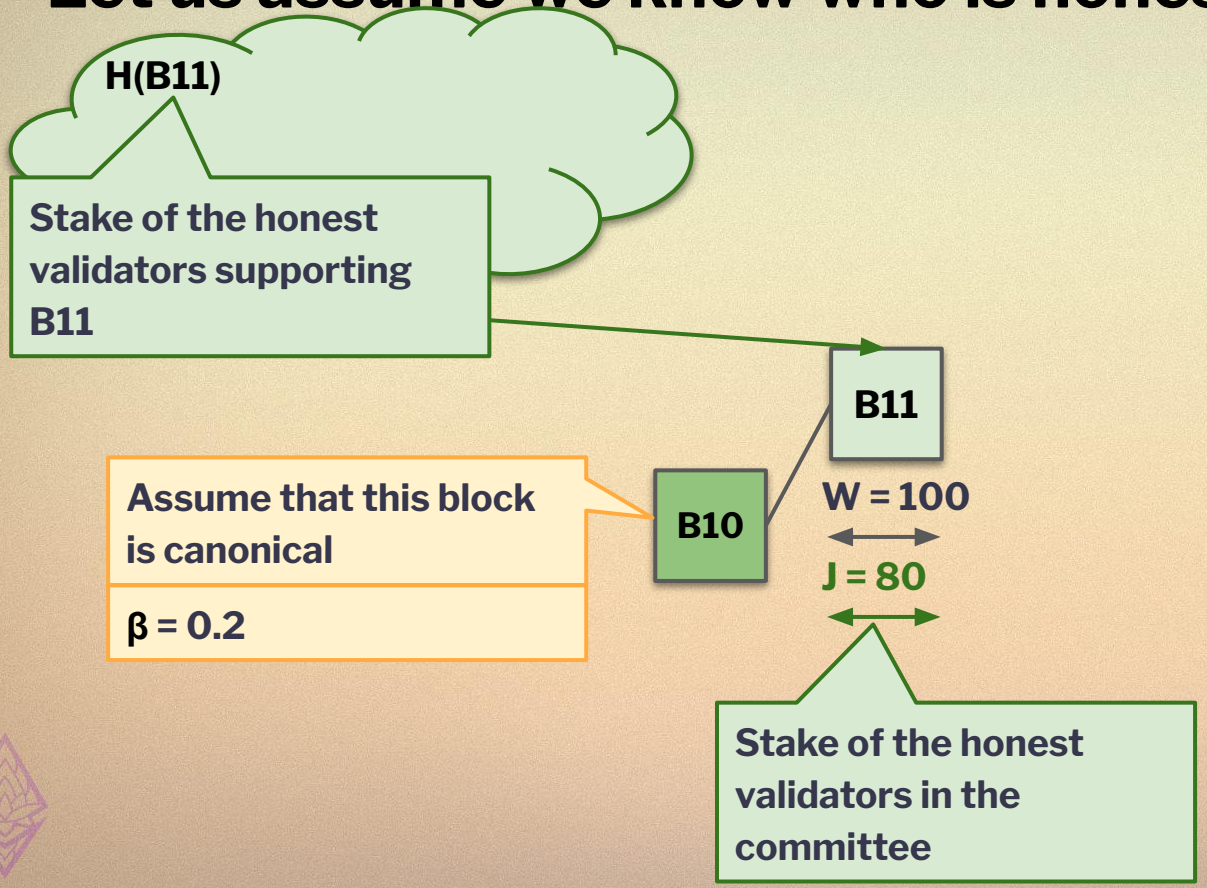
What about when we move to the next slot?



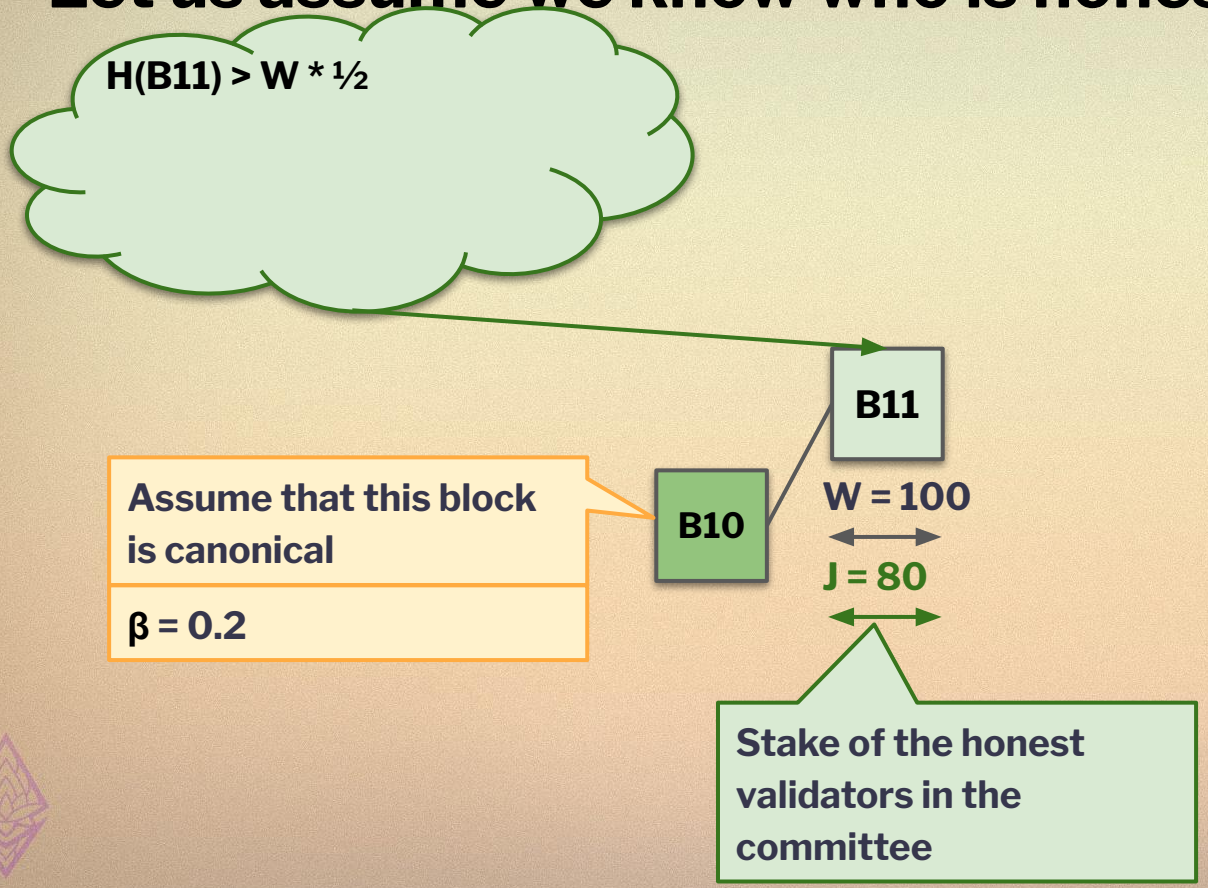
Let us assume we know who is honest



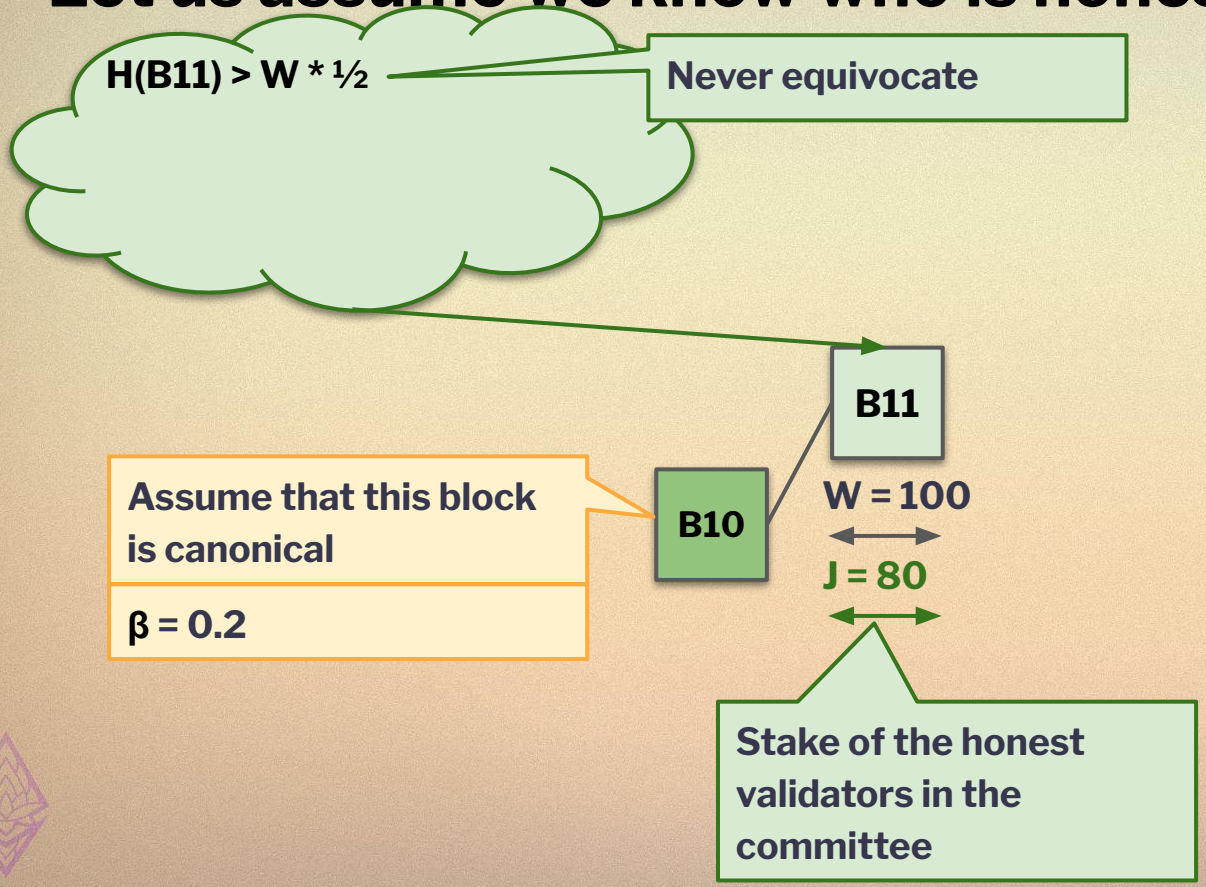
Let us assume we know who is honest



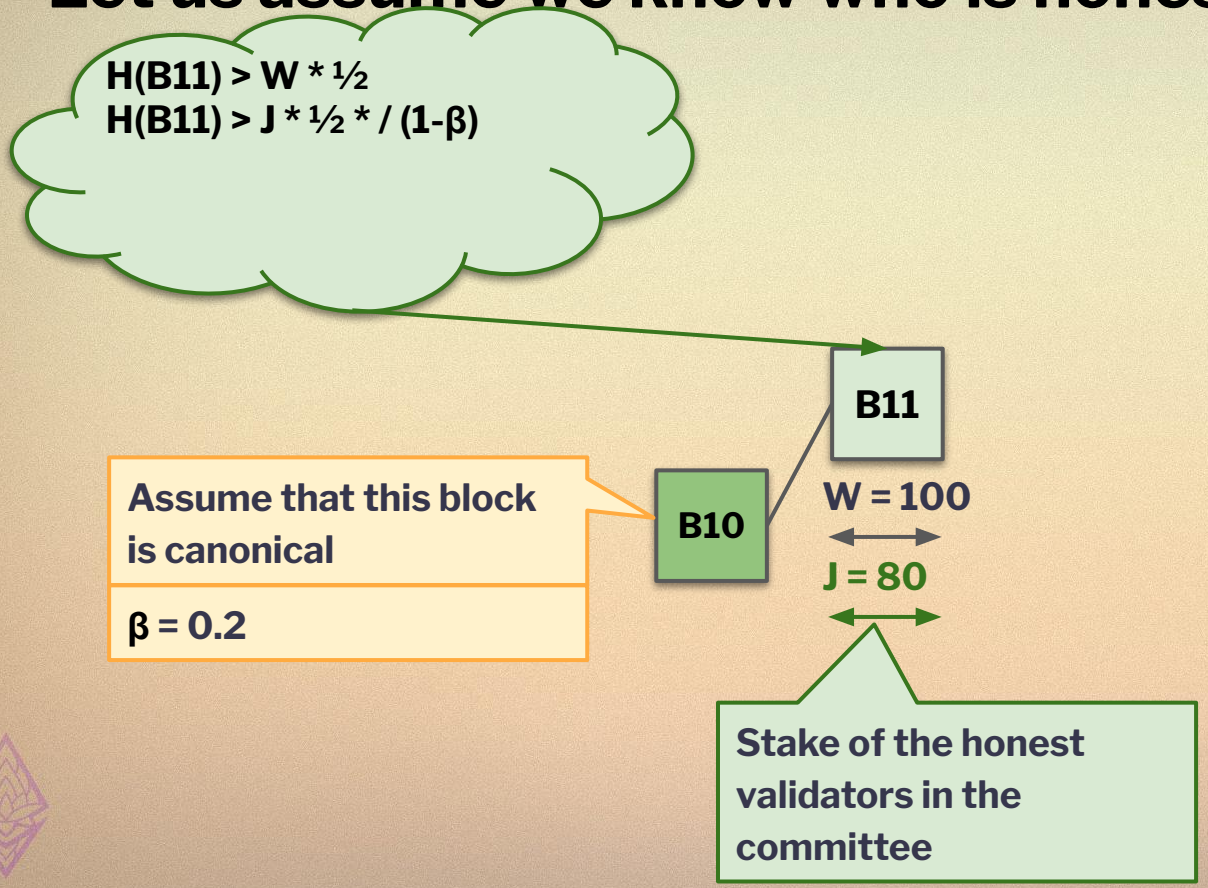
Let us assume we know who is honest



Let us assume we know who is honest



Let us assume we know who is honest



Let us assume we know who i

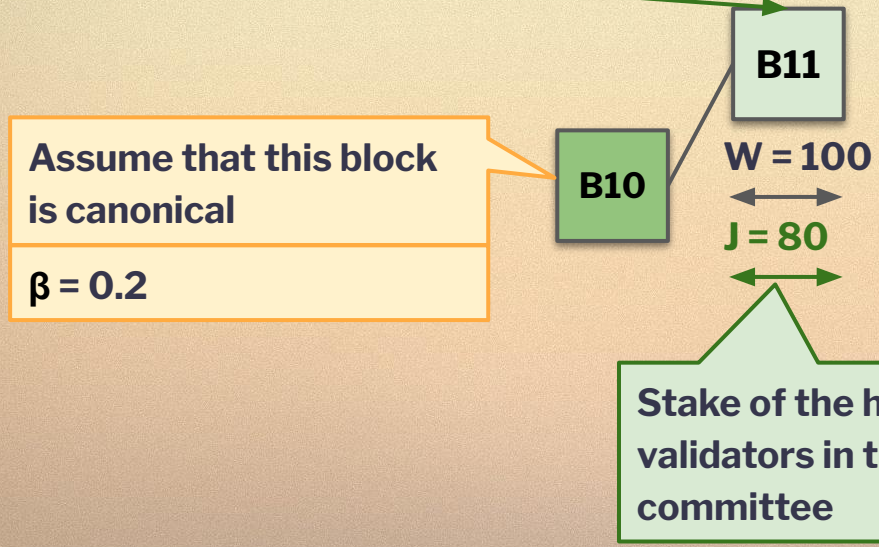
$J \geq (1-\beta) * W$

$J / (1-\beta) \geq W$

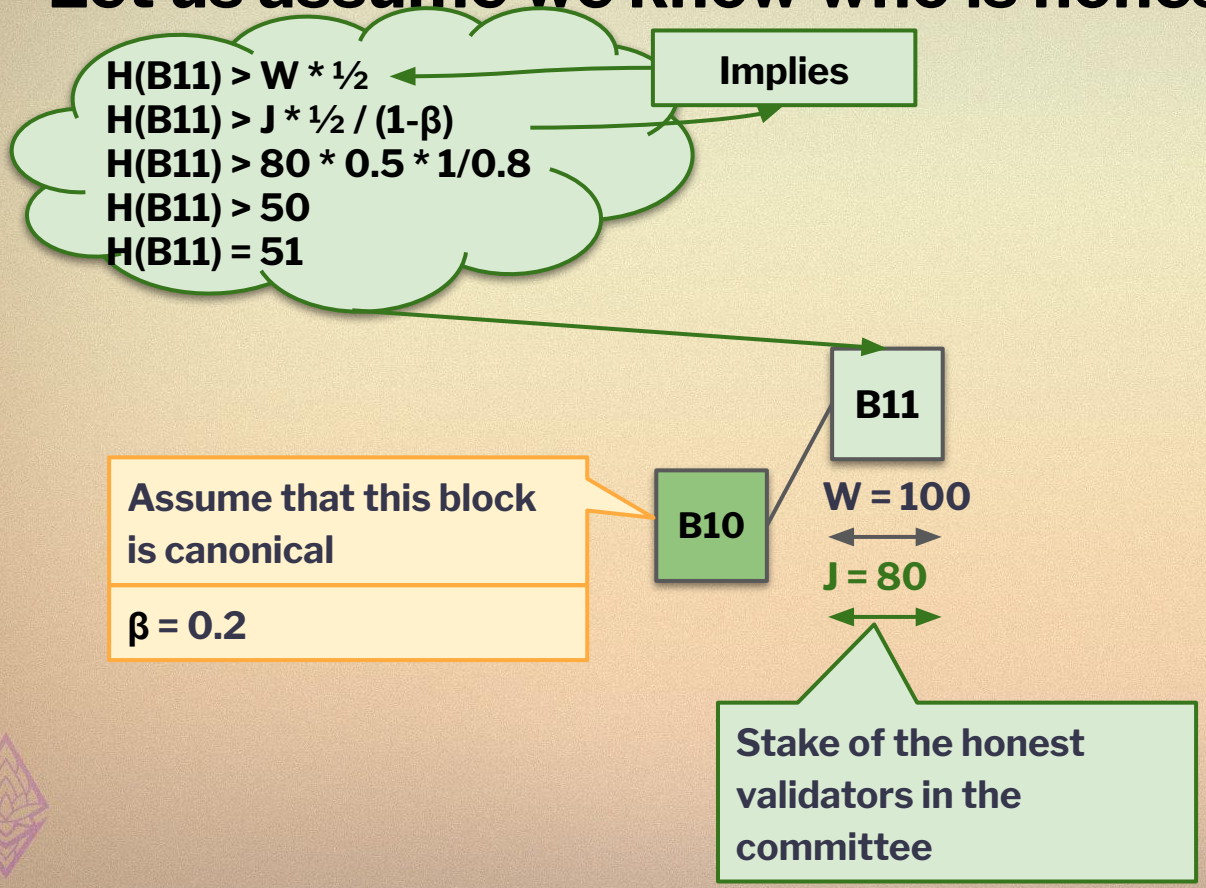
Implies

$H(B11) > W * ½$

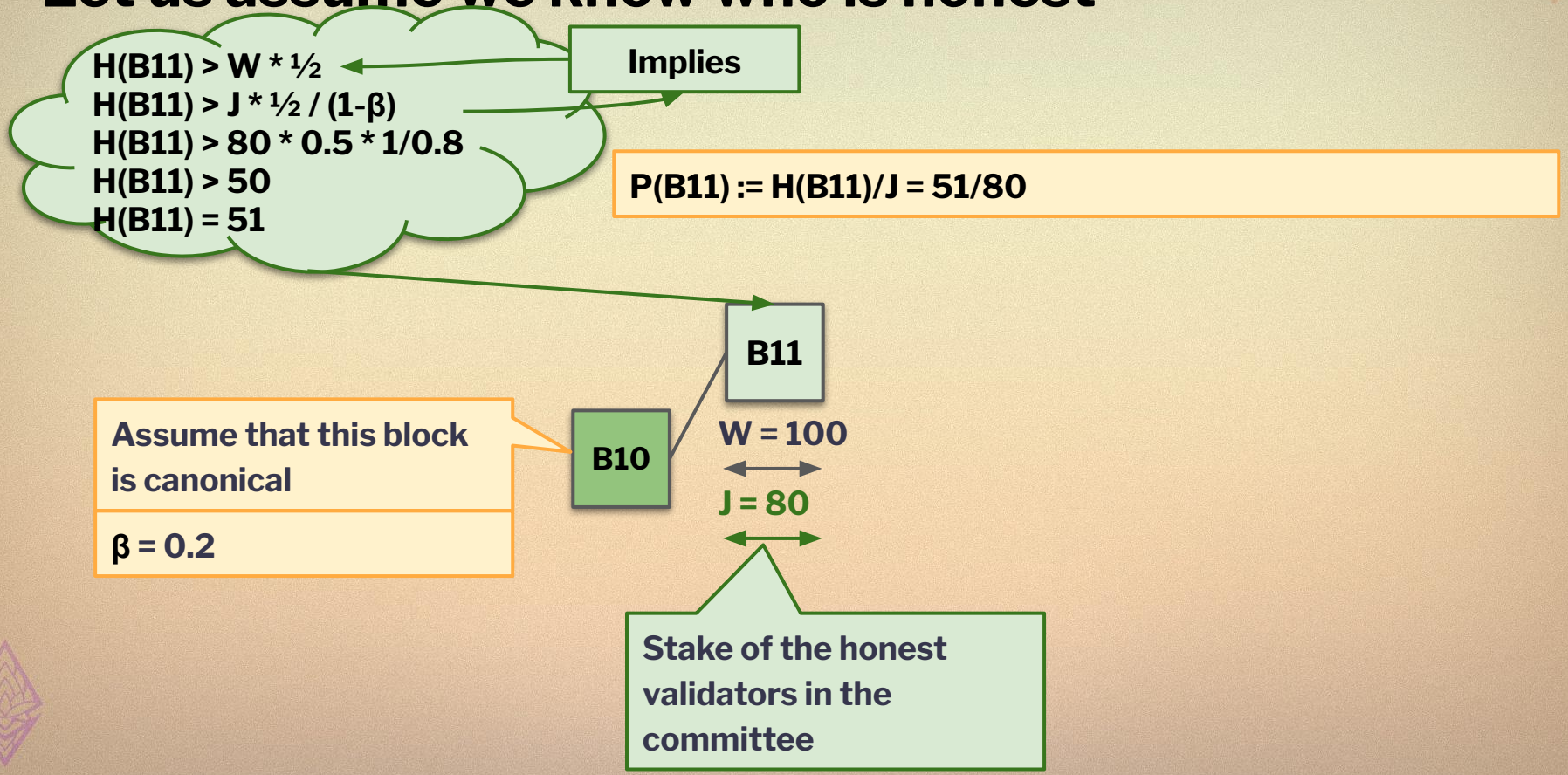
$H(B11) > J * ½ * / (1-\beta)$



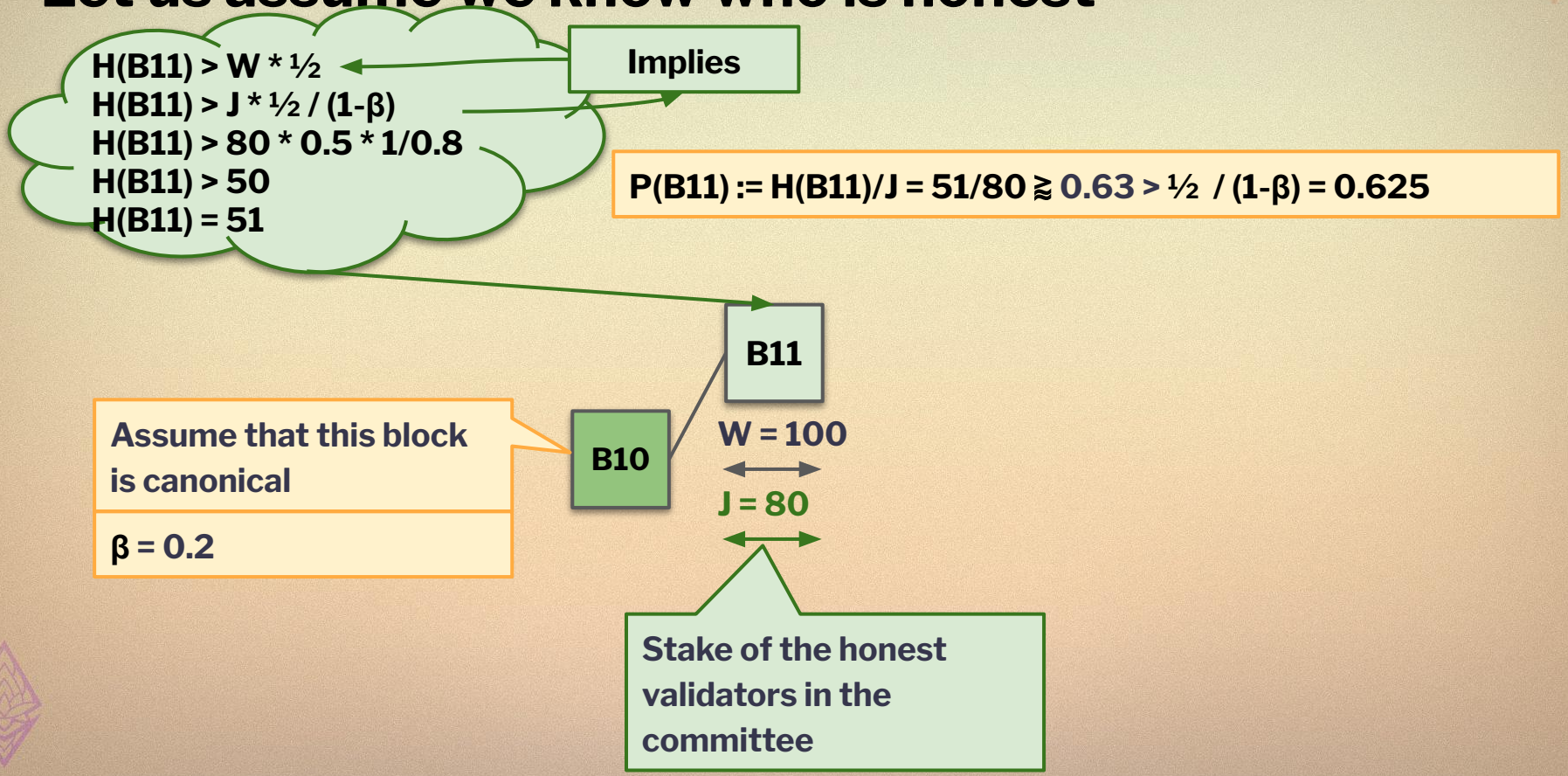
Let us assume we know who is honest



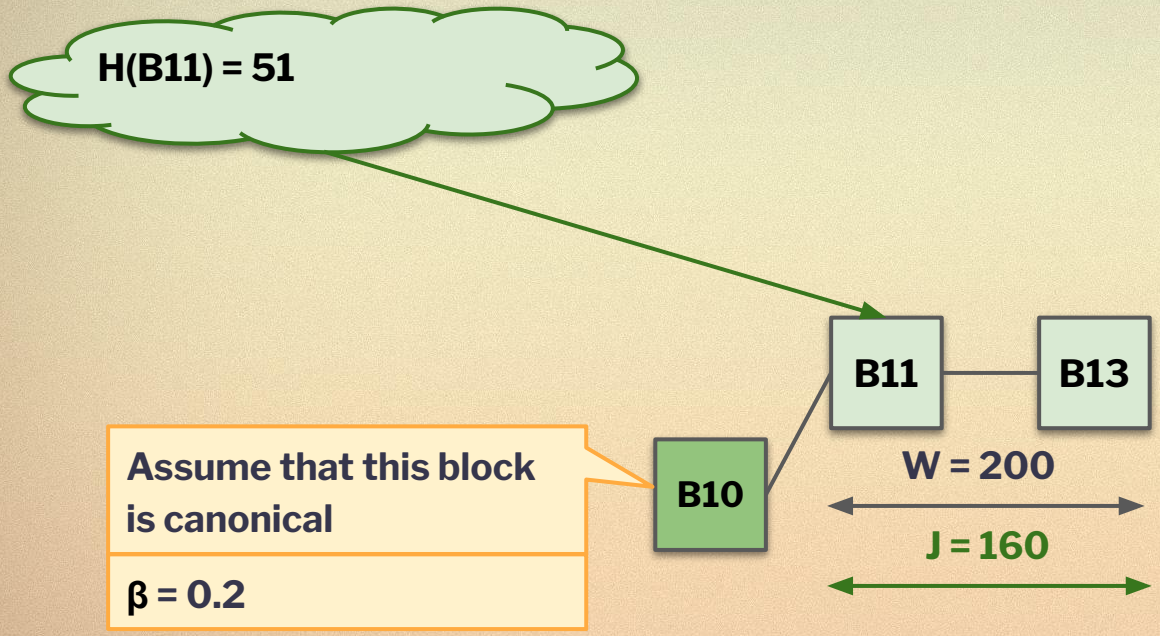
Let us assume we know who is honest



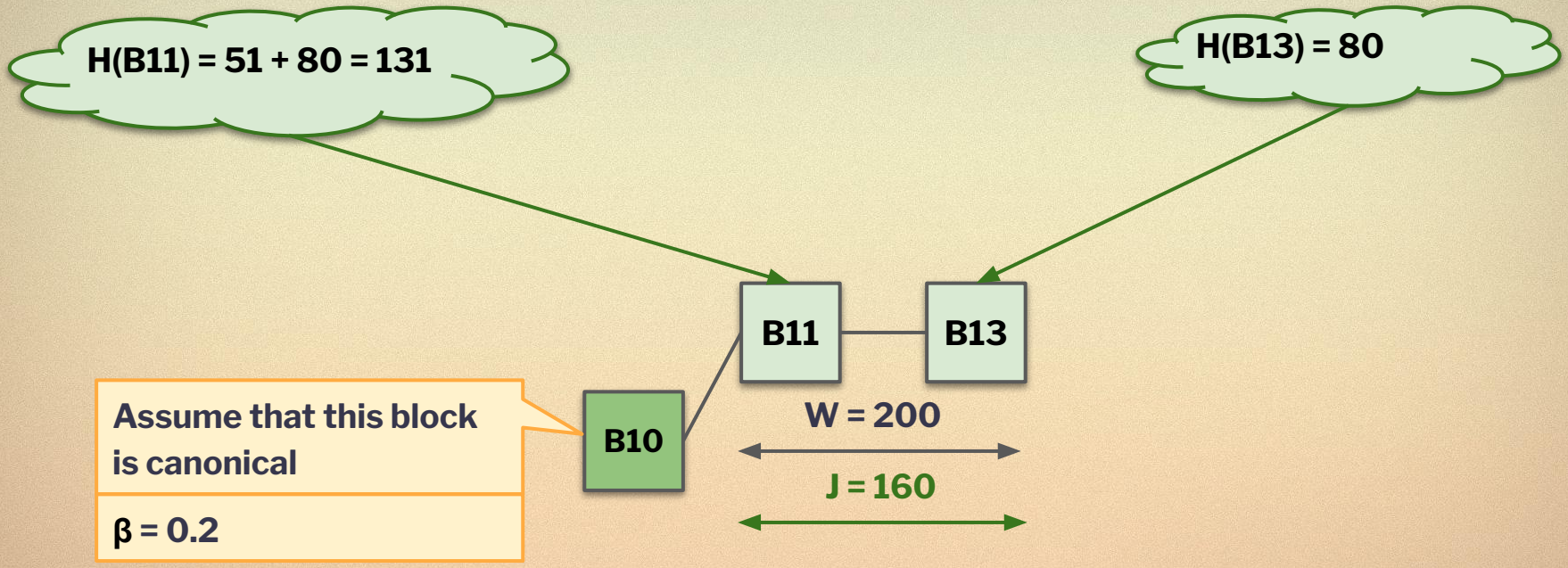
Let us assume we know who is honest



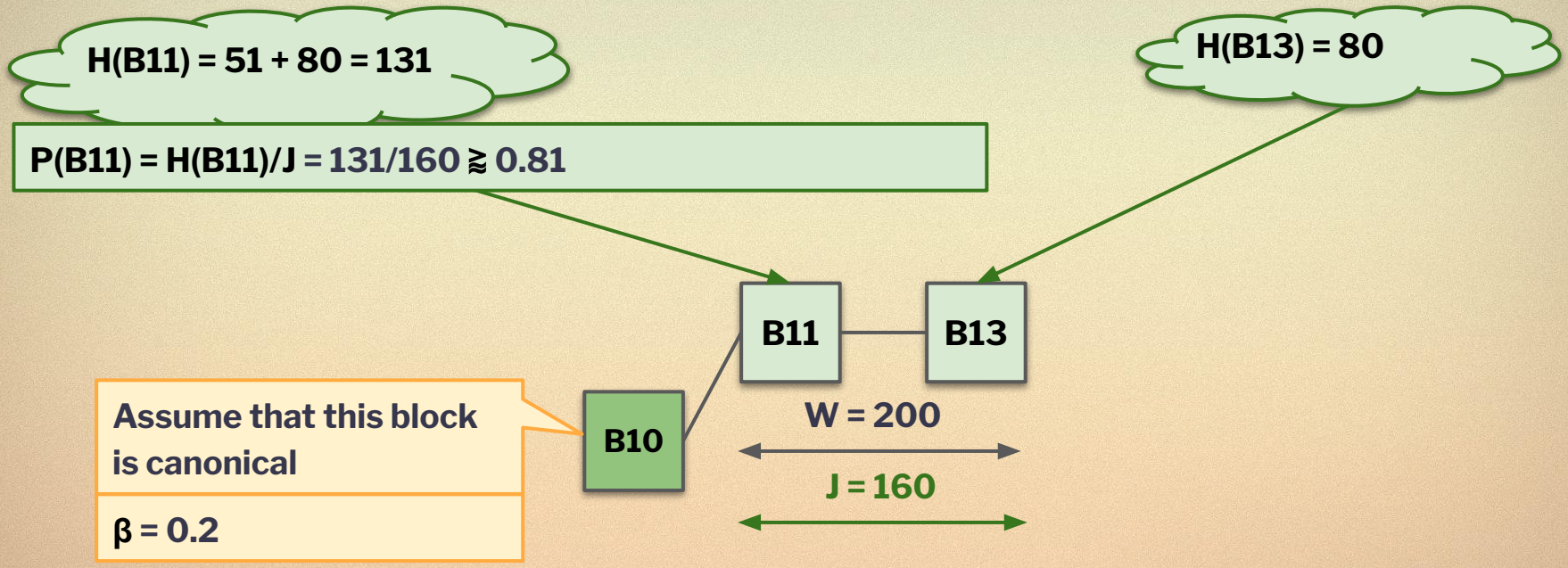
What happens at the next slot?



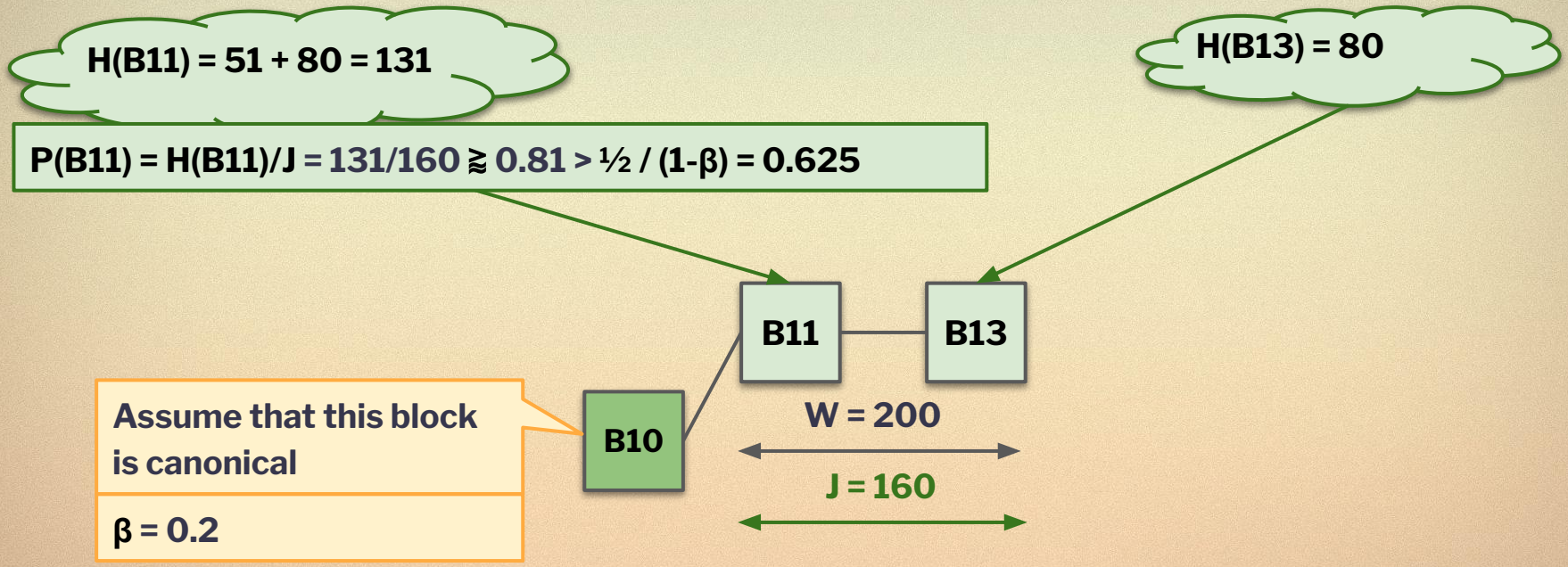
What happens at the next slot?



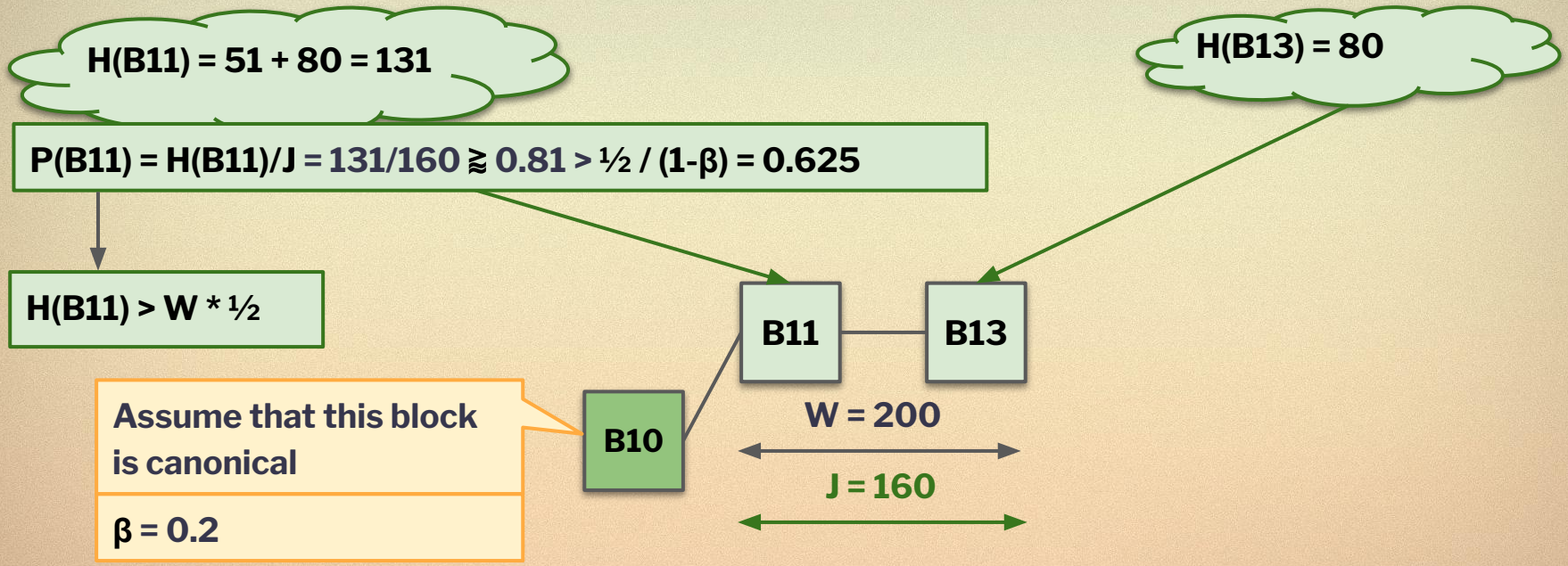
What happens at the next slot?



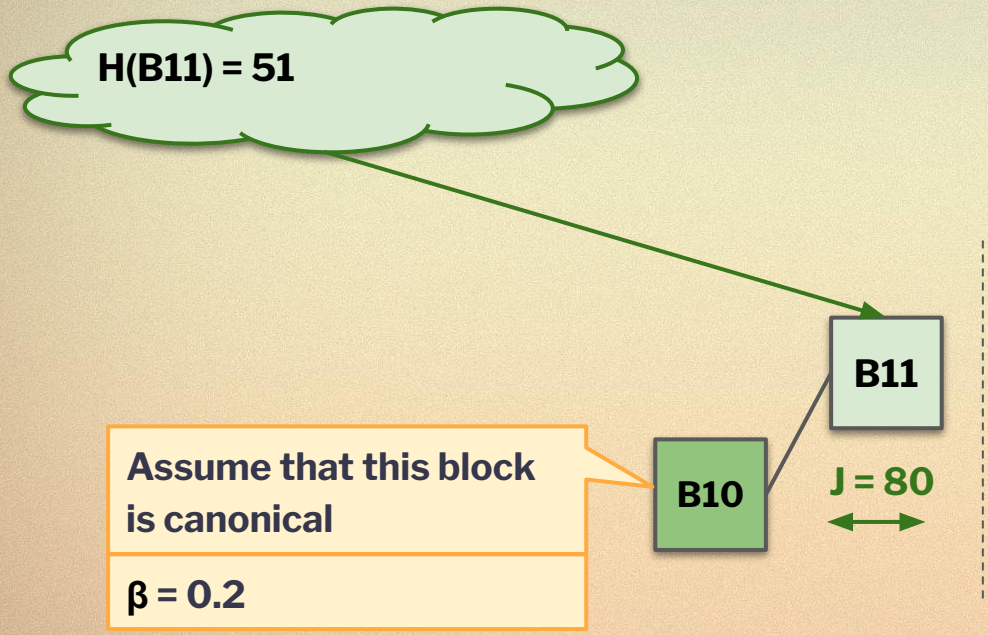
What happens at the next slot?



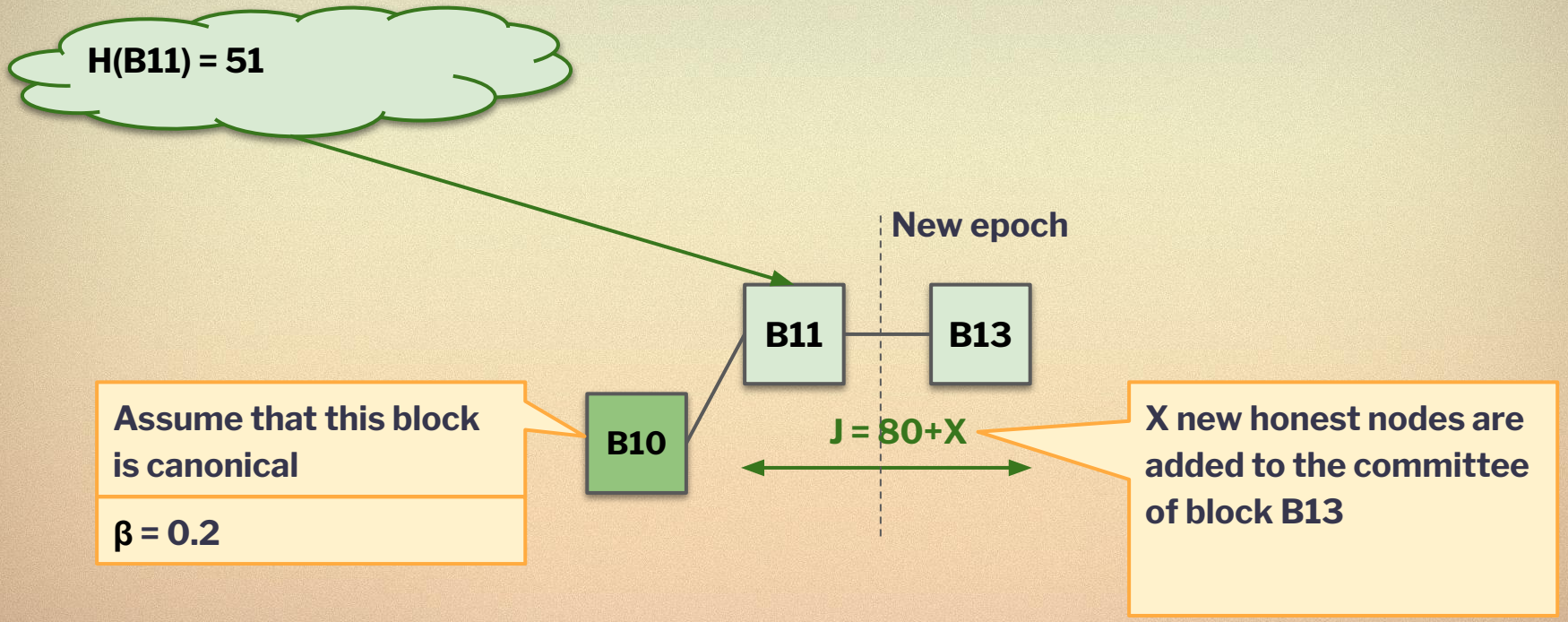
What happens at the next slot?



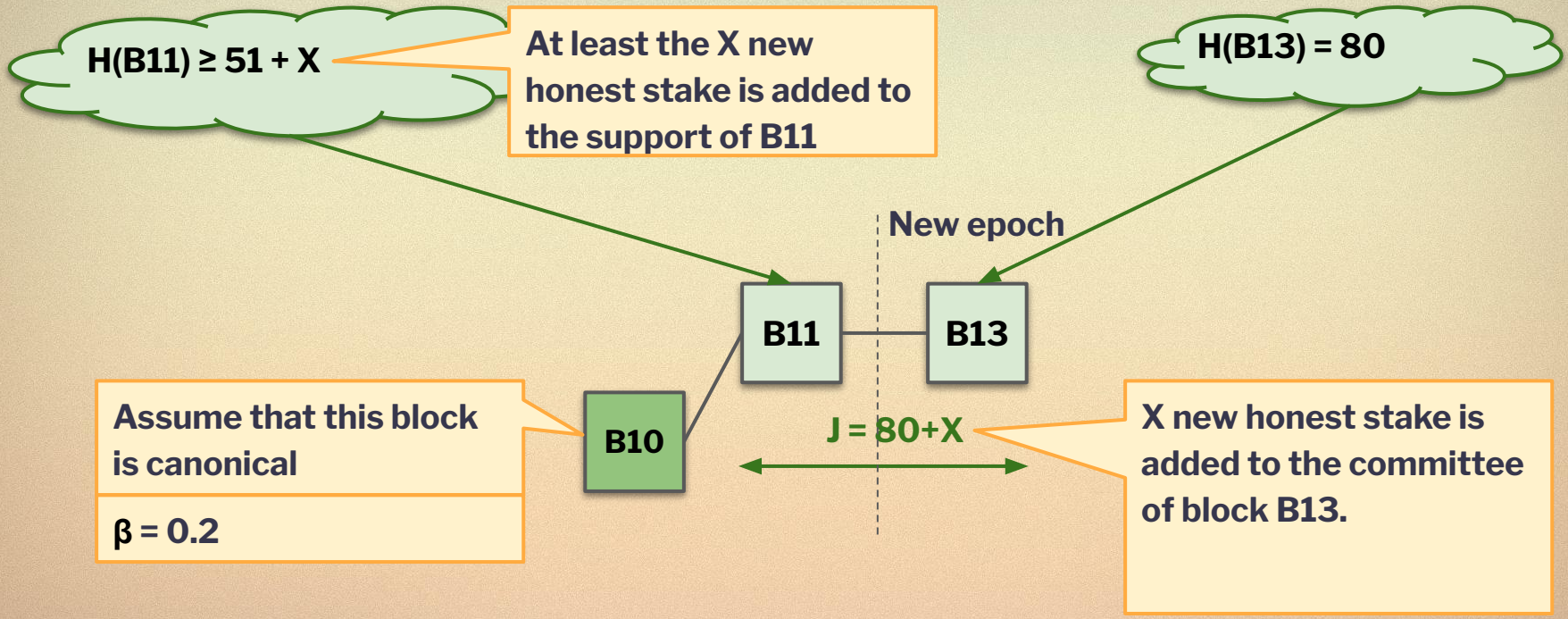
What happens if the next slot is in a new epoch?



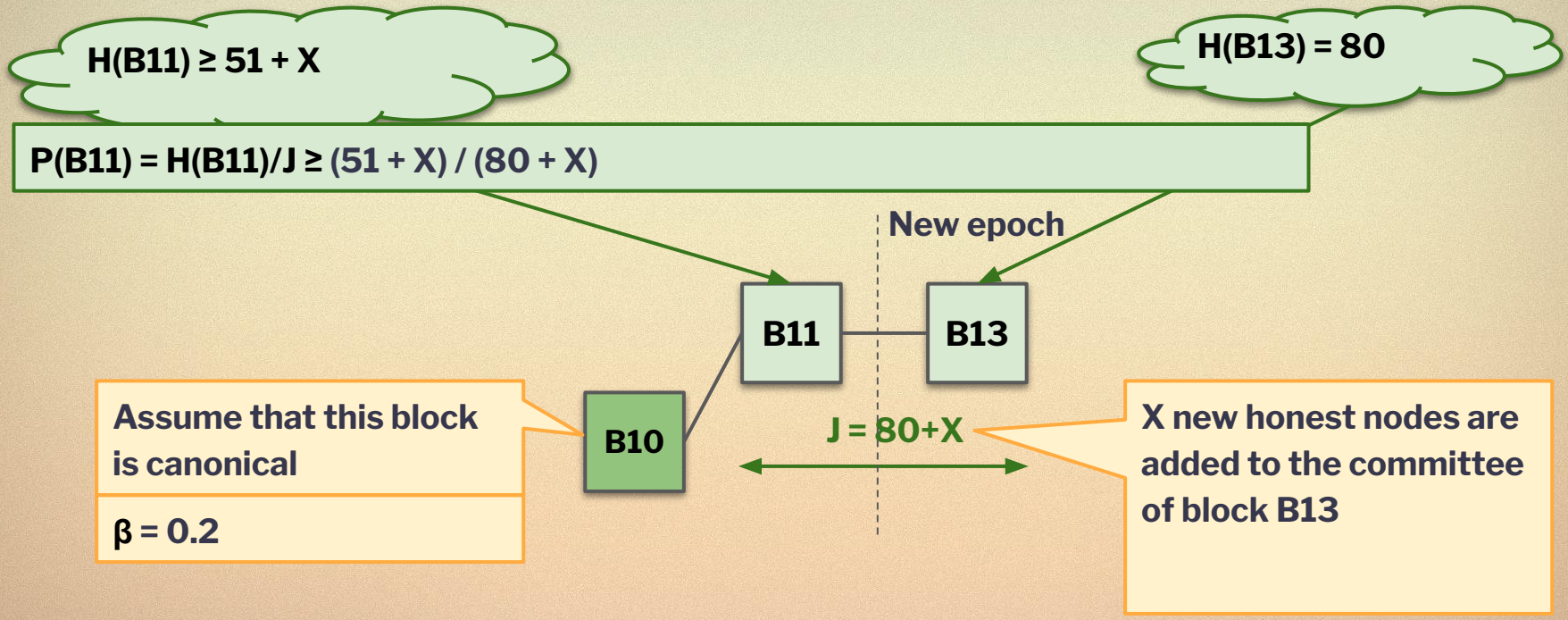
What happens if the next slot is in a new epoch?



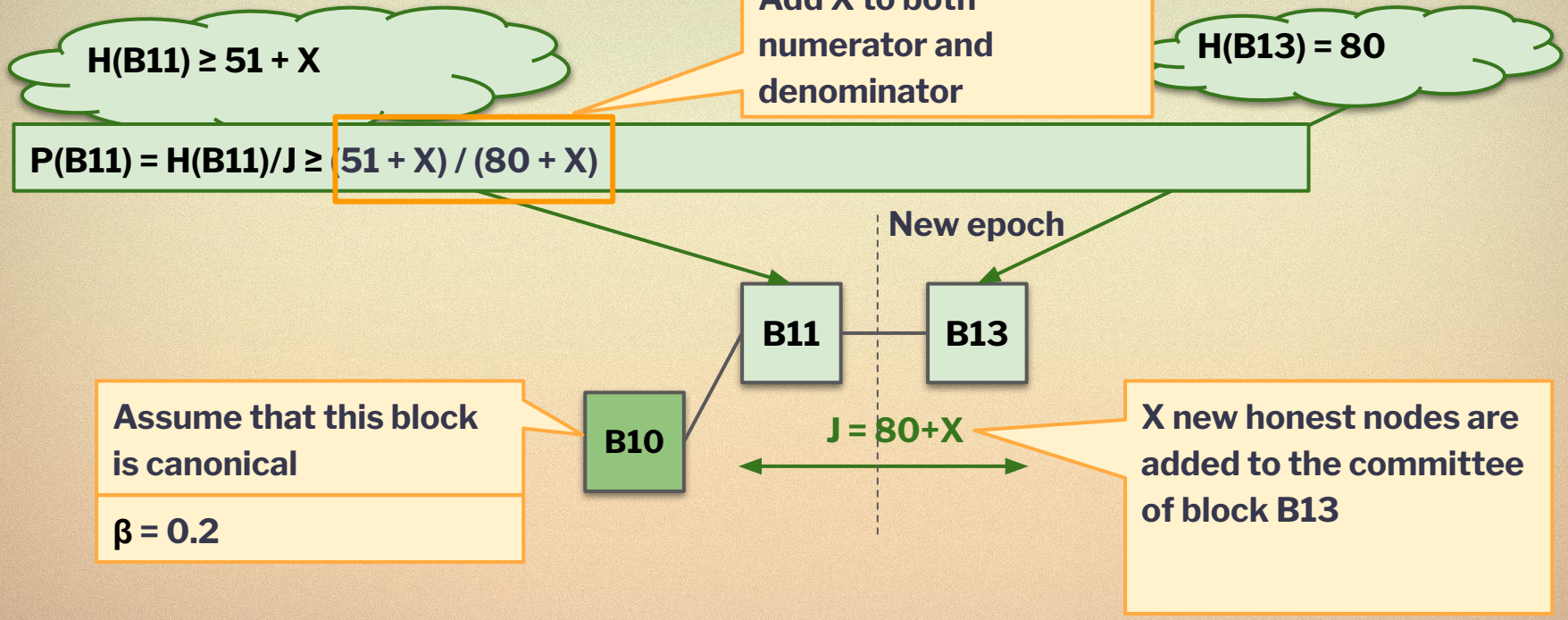
What happens if the next slot is in a new epoch?



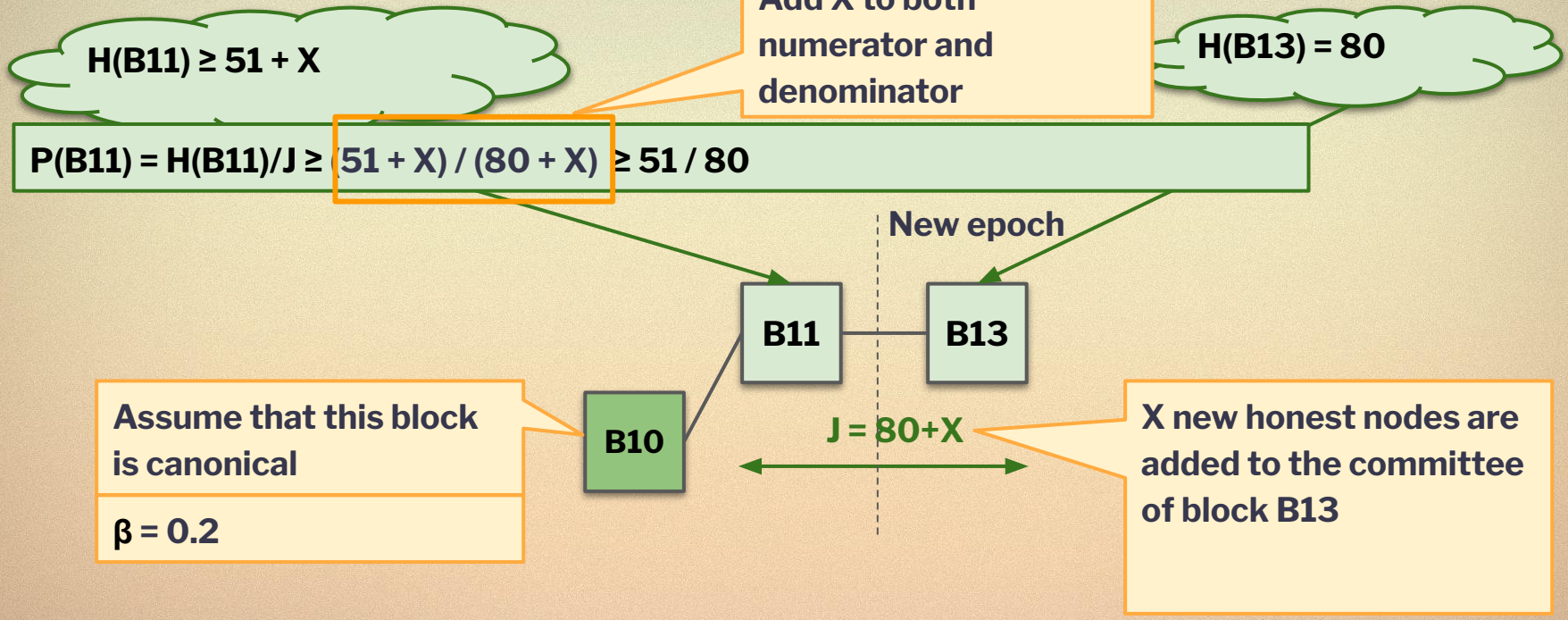
What happens if the next slot is in a new epoch?



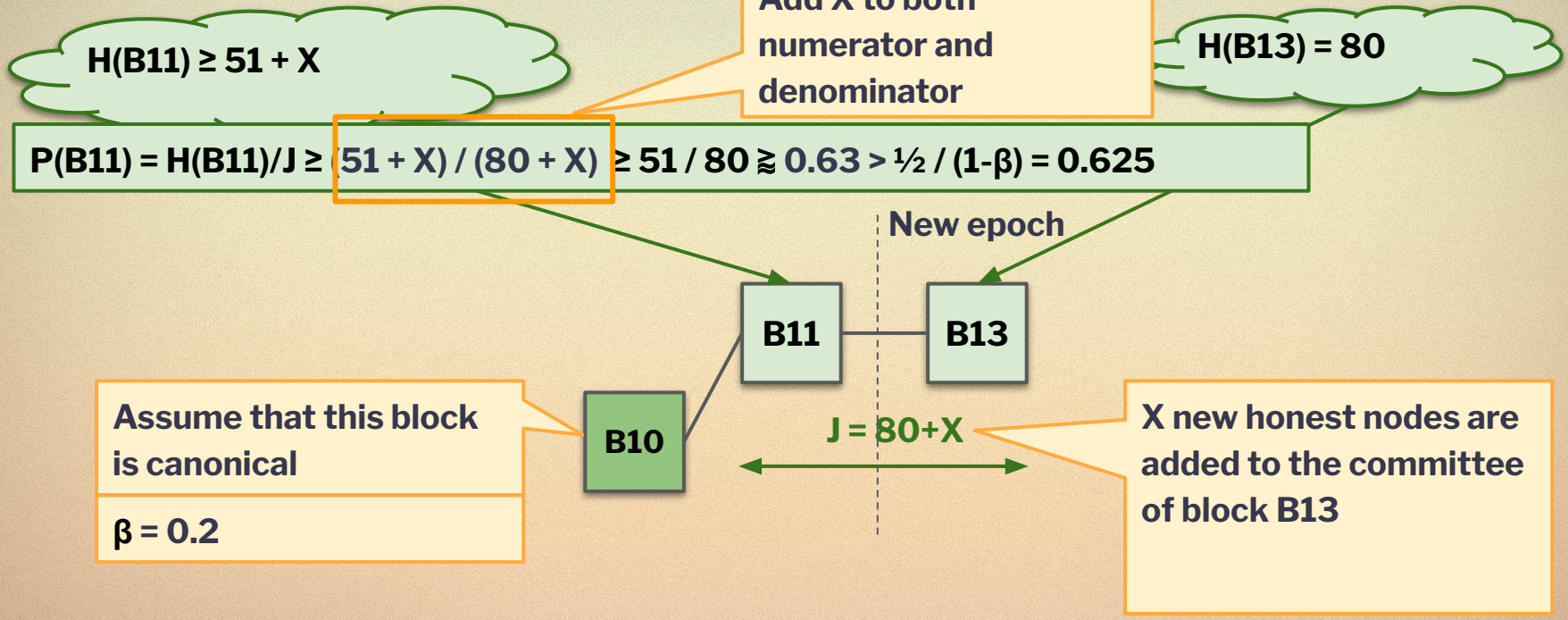
What happens if the next slot is in a new epoch?



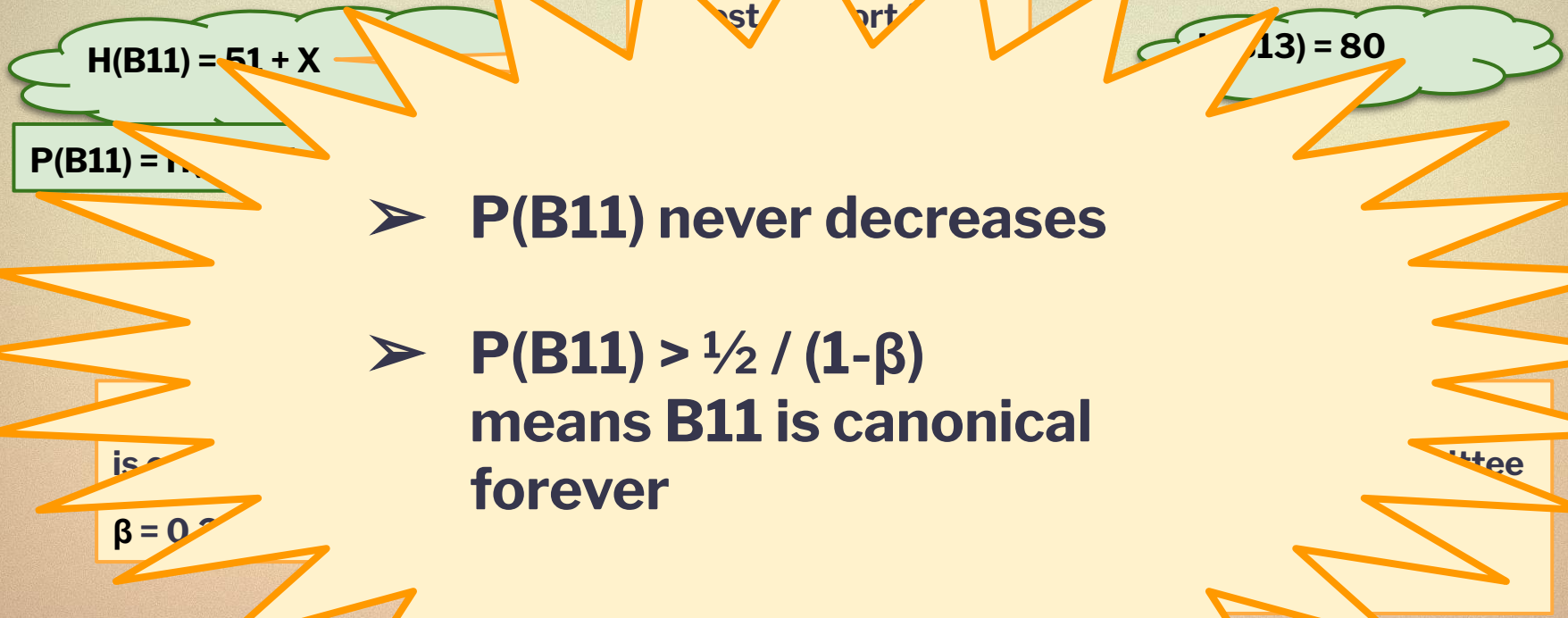
What happens if the next slot is in a new epoch?



What happens if the next slot is in a new epoch?



What happens if the next slot is in a new epoch?



How do we measure P(B11)

How do we measure P(B11)

$$S(B11) / W > \frac{1}{2} + \beta$$

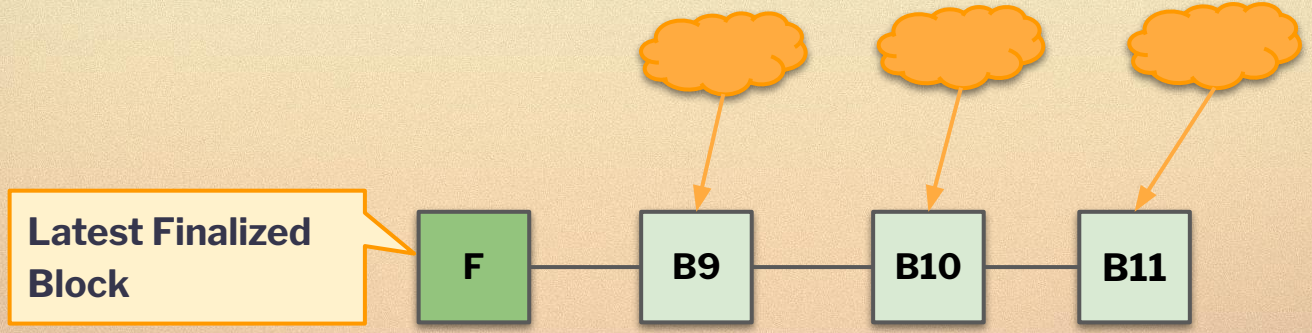
Implies

$$P(B11) = H(B11) / J > \frac{1}{2} / (1-\beta)$$

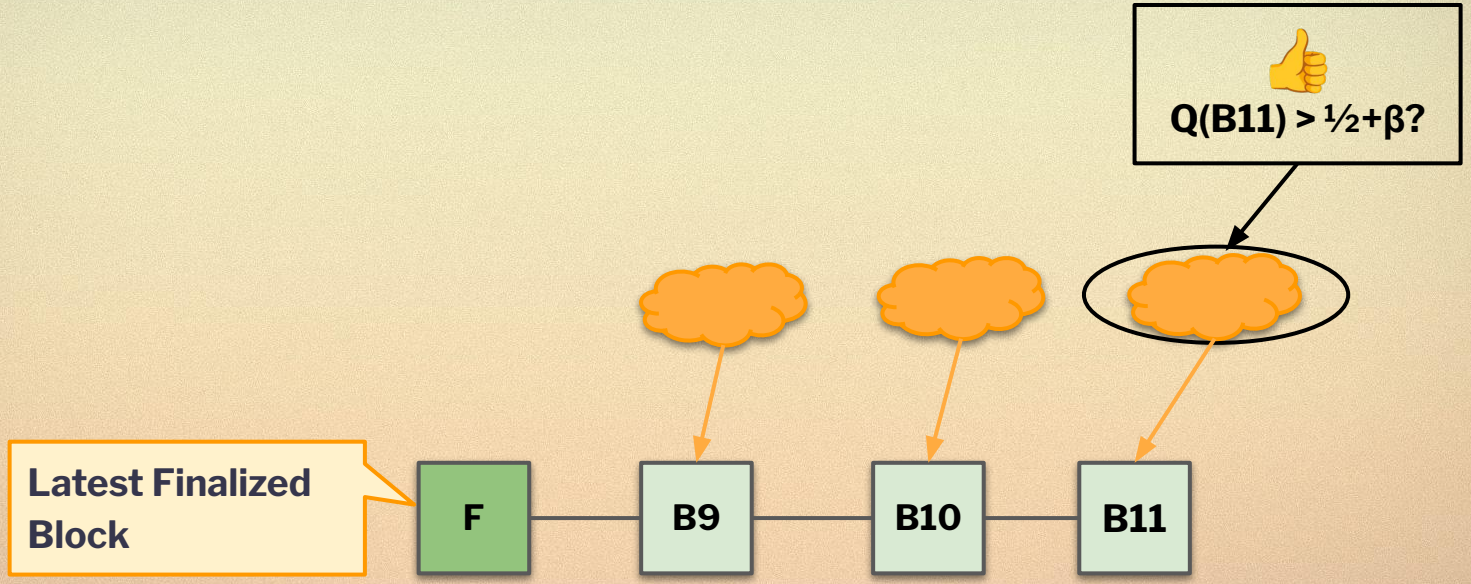
How do we measure P(B11)

- $Q(B11) := S(B11)/W > \frac{1}{2} + \beta$
- implies $P(B11) > \frac{1}{2} / (1-\beta)$
- which means B11 is canonical forever

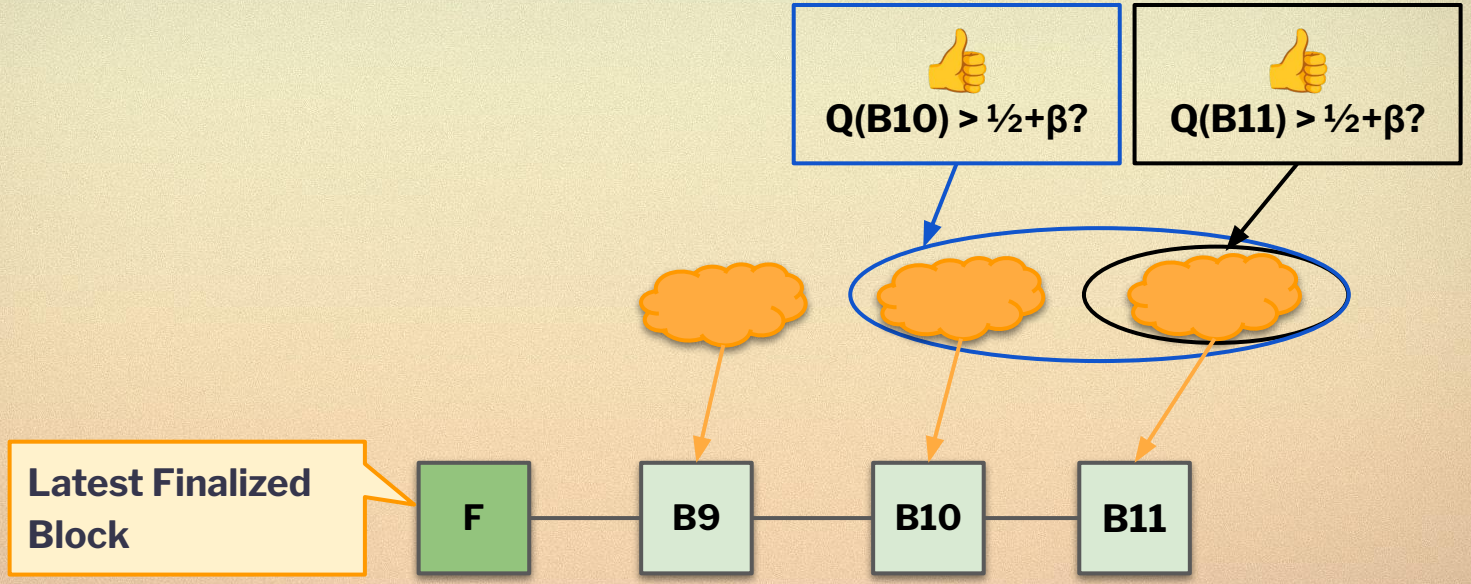
What about B10 and the other blocks?



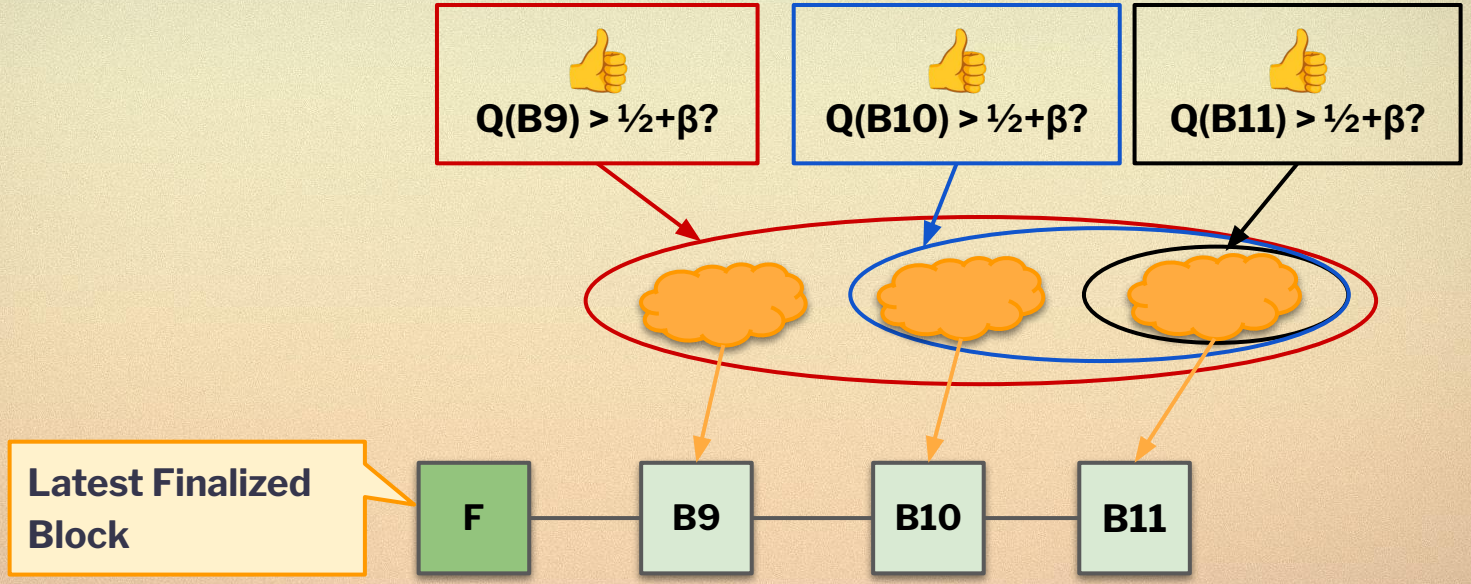
What about B10 and the other blocks?



What about B10 and the other blocks?



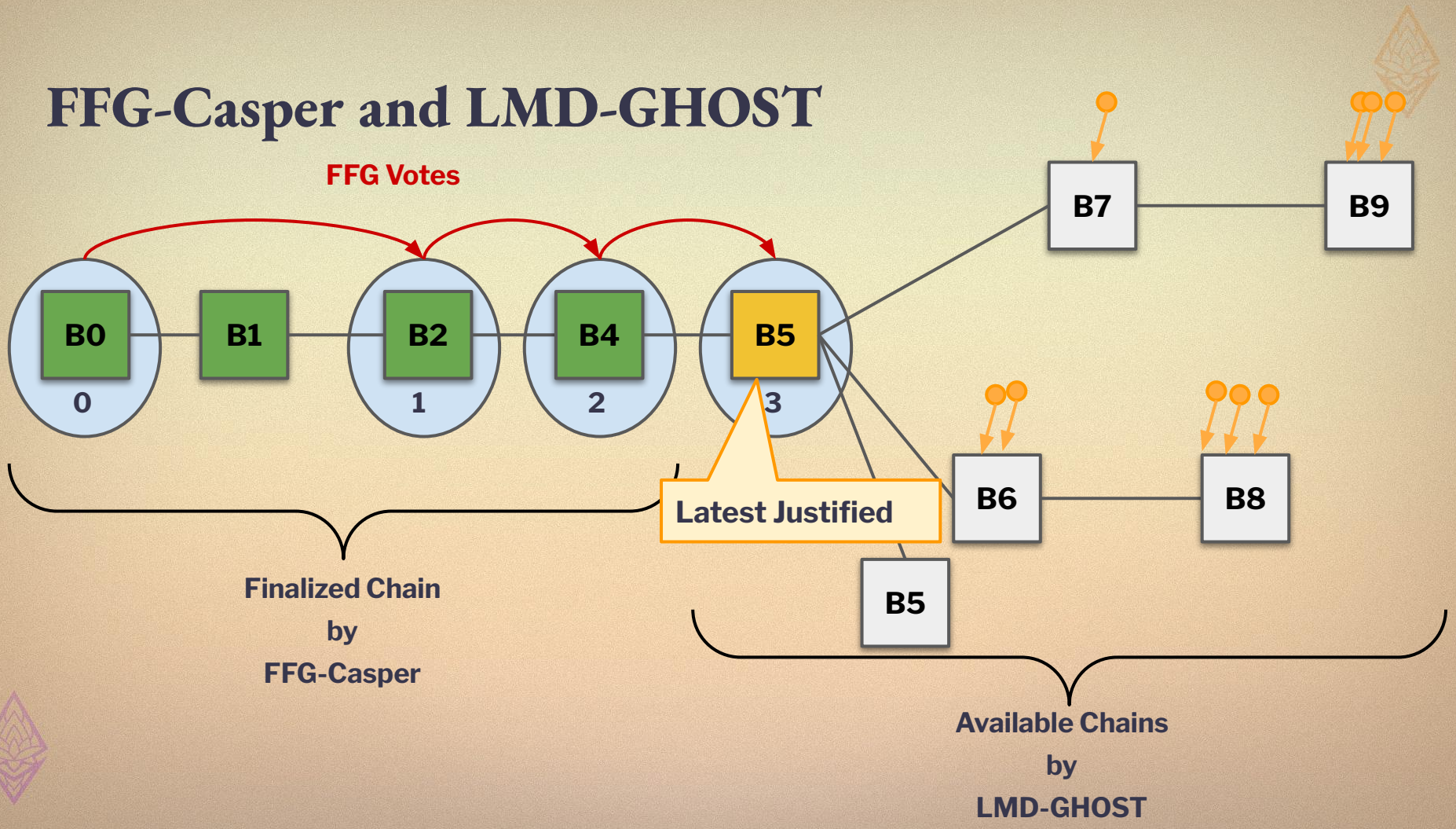
What about B10 and the other blocks?



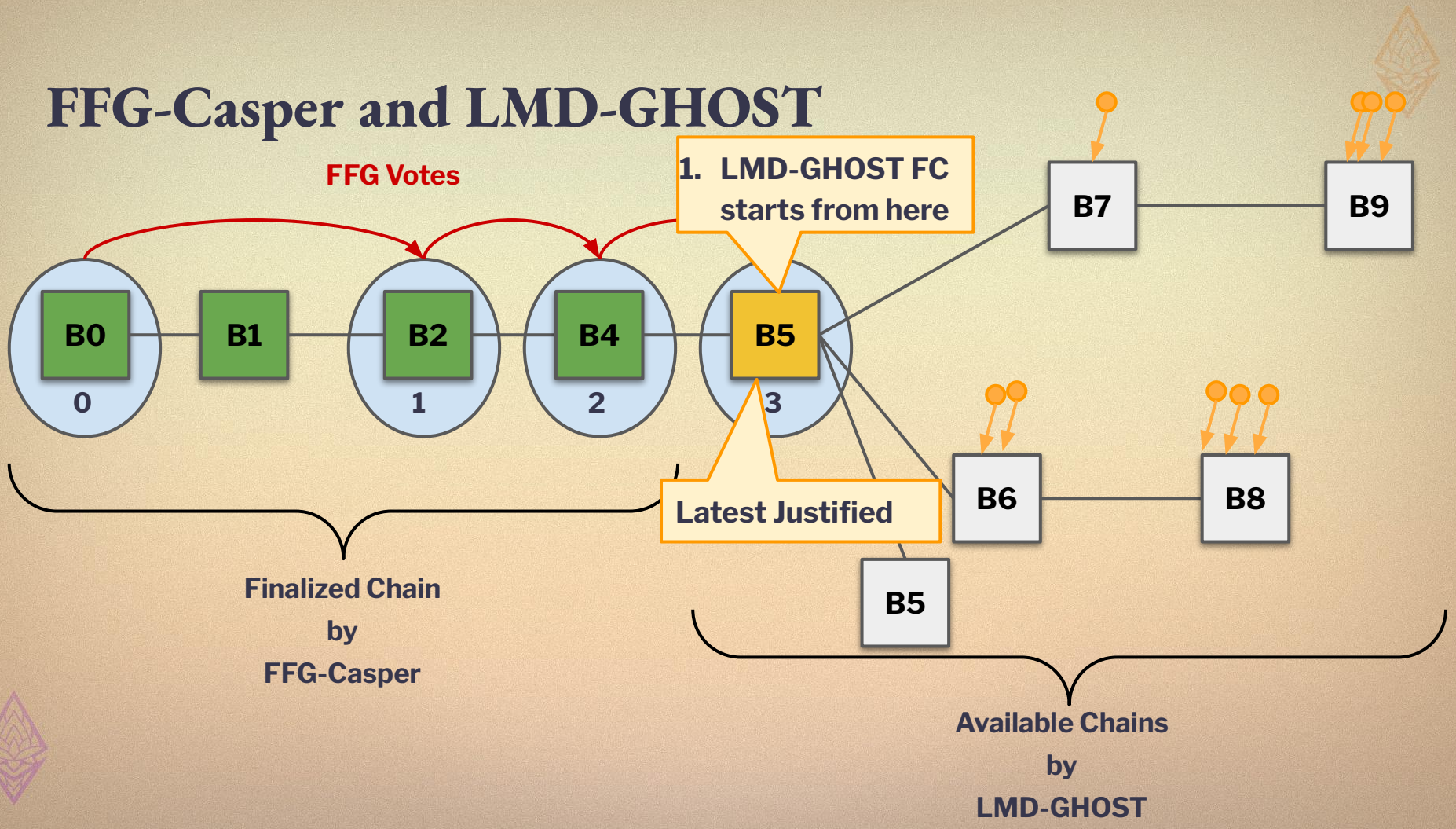
Section 5

Adding FFG Casper

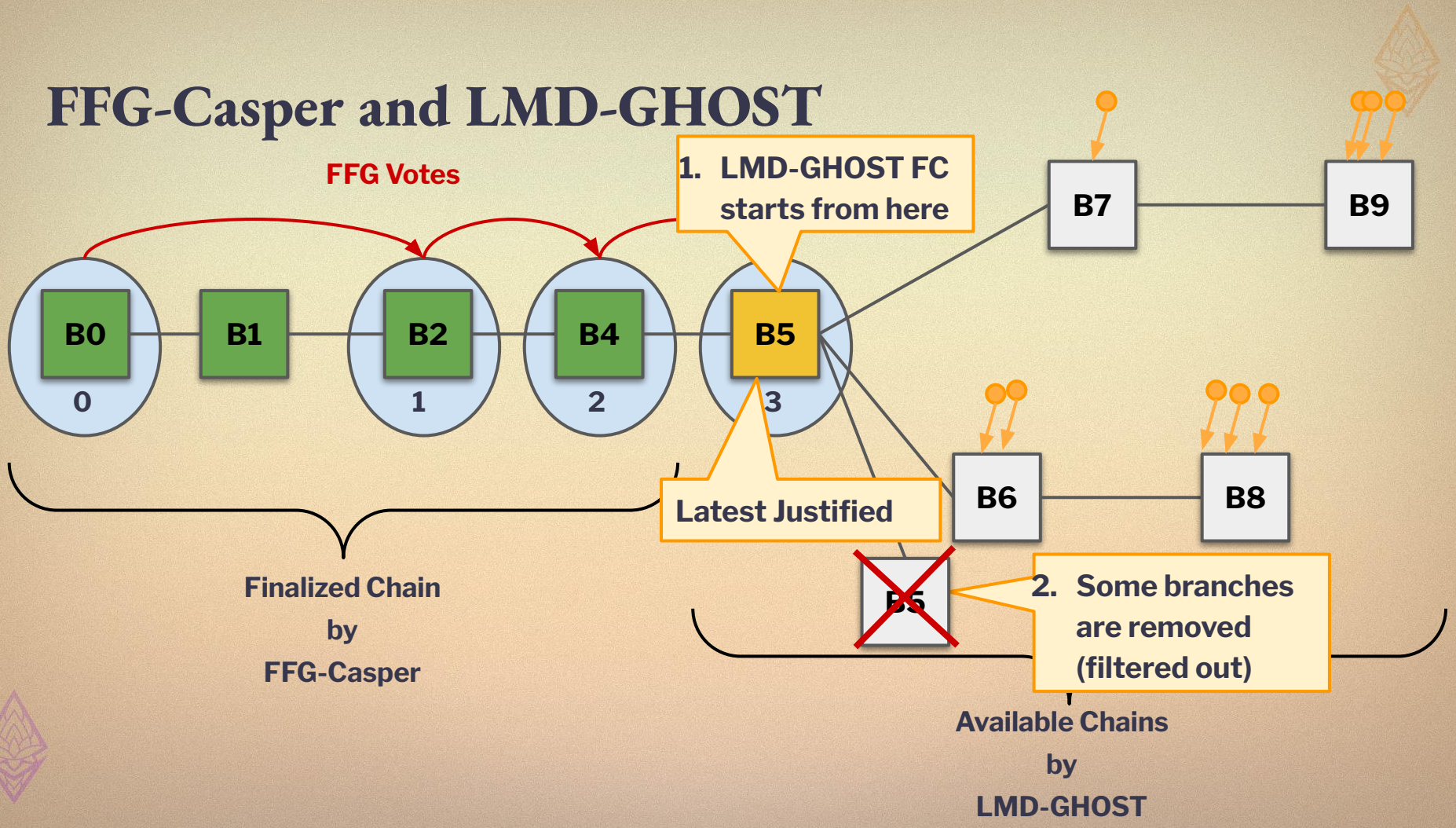
FFG-Casper and LMD-GHOST



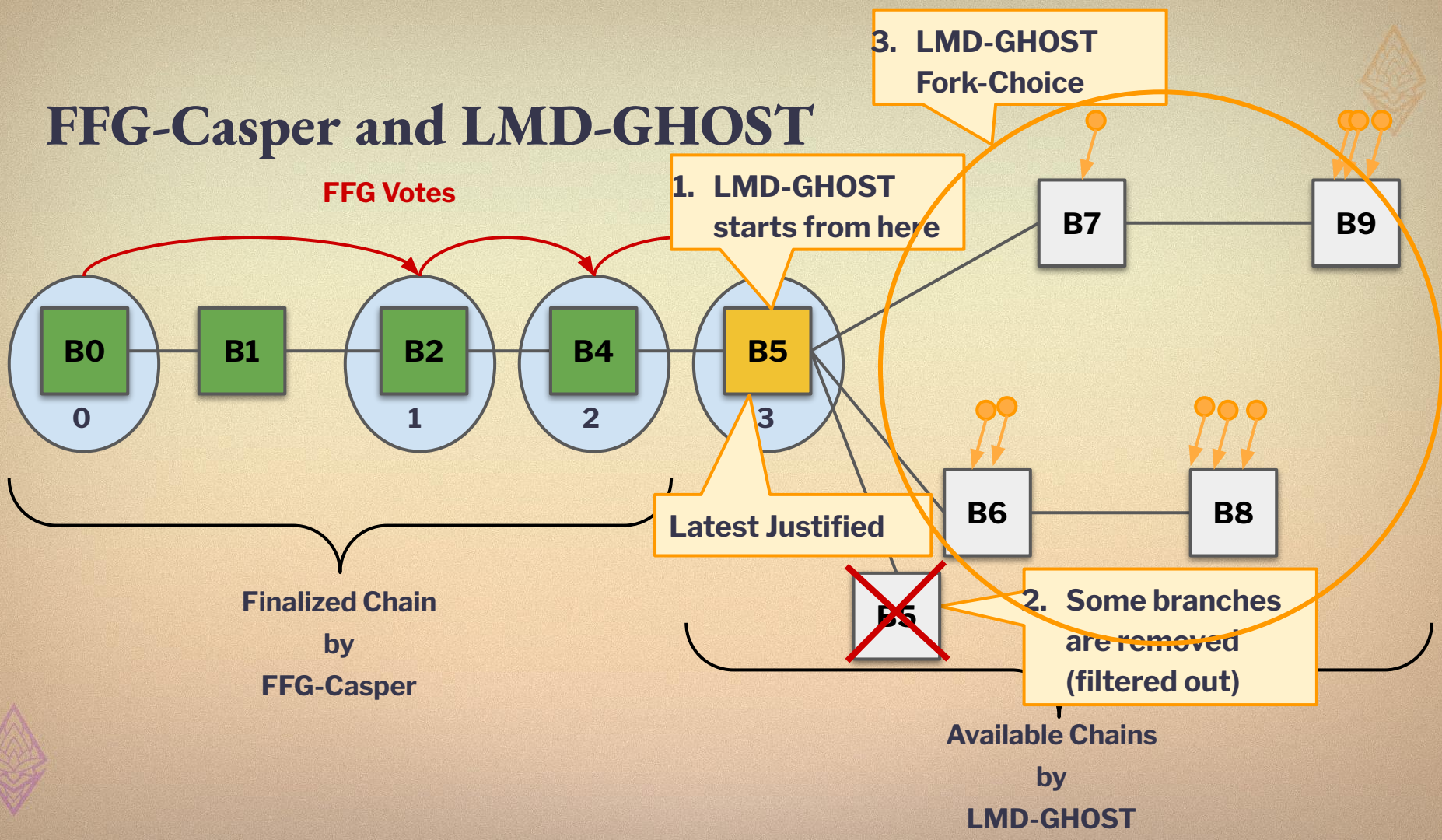
FFG-Casper and LMD-GHOST



FFG-Casper and LMD-GHOST



FFG-Casper and LMD-GHOST



Final rule

- **isLMDConfirmed(B)**

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG
  - epoch(Justified(B)) = current_epoch - 1

B is not filtered out in the current epoch

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG
  - epoch(Justified(B)) = current_epoch - 1
  - FFG_votes(checkpoint(B))
    - β * committee_stake ([first_slot_epoch, ..., current_slot])
    + (1-β) * committee_stake ([current_slot+1, ..., last_slot_epoch])
    ≥ ⅔ total_stake

B is not filtered out in the current epoch

B is not filtered out in any future epoch

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG
  - epoch(Justified(B)) = current_epoch - 1
  - FFG_votes(checkpoint(B))
    - β * committee_stake ([first_slot_epoch, ..., current_slot])
    + (1-β) * committee_stake ([current_slot+1, ..., last_slot_epoch])
    ≥ ⅔ total_stake

Minimum stake that FFG votes for checkpoint(B) up until the current slot

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG
  - epoch(Justified(B)) = current_epoch - 1
  - FFG_votes(checkpoint(B))
    - β * committee_stake ([first_slot_epoch, ..., current_slot])
    + (1-β) * committee_stake ([current_slot+1, ..., last_slot_epoch])
    ≥ ⅔ total_stake

Minimum stake that FFG votes for checkpoint(B) up until the current slot

Minimum stake that FFG votes for checkpoint(B) from next slot till the end of the epoch

Final rule

- isLMDConfirmed(B)
- B is not filtered out when considering Casper-FFG
  - epoch(Justified(B)) = current_epoch - 1
  - FFG_votes(checkpoint(B))
    - β * committee_stake ([first_slot_epoch, ..., current_slot])
    + (1-β) * committee_stake ([current_slot+1, ..., last_slot_epoch])
    ≥ ⅔ total_stake

Minimum stake that FFG votes for checkpoint(B) up until the current slot

Minimum required FFG stake to justify checkpoint(B)

Minimum stake that FFG votes for checkpoint(B) from next slot till the end of the epoch

Section 6

Initial Experimental Results

Initial Experimental Results by

Blog post coming soon. Stay tuned!

Setup

- Run the Confirmation Rule over 6 days
- 56 blocks were detected as re-orged, including a 2-block re-org.
- Confirmation Rule executed by polling a Beacon Node at 10s intervals, rather than executing it at the beginning of each slot
- FFG weight estimated from LDM weight

Initial Experimental Results by

Setup

- Run the Confirmation Rule over 6 days
- 56 blocks were detected as re-orged, including a 2-block re-org.
- Confirmation Rule executed by polling a Beacon Node at 10s intervals, rather than executing it at the beginning of each slot
- FFG weight estimated from LDM weight

Results

- **None of the re-orged blocks were confirmed under the rule, no matter how the adversarial thresholds were tuned.**

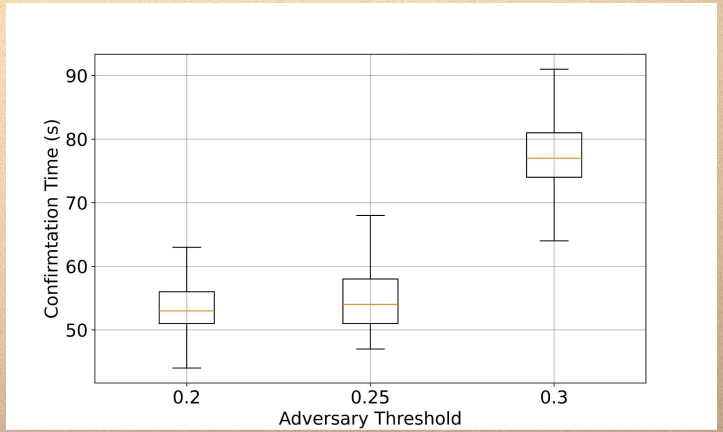
Initial Experimental Results by CIRCLE

Setup

- Run the Confirmation Rule over 6 days
- 56 blocks were detected as re-orged, including a 2-block re-org.
- Confirmation Rule executed by polling a Beacon Node at 10s intervals, rather than executing it at the beginning of each slot
- FFG weight estimated from LDM weight

Results

- **None of the re-orged blocks were confirmed under the rule, no matter how the adversarial thresholds were tuned.**



Initial Experimental Results by CIRCLE

Setup

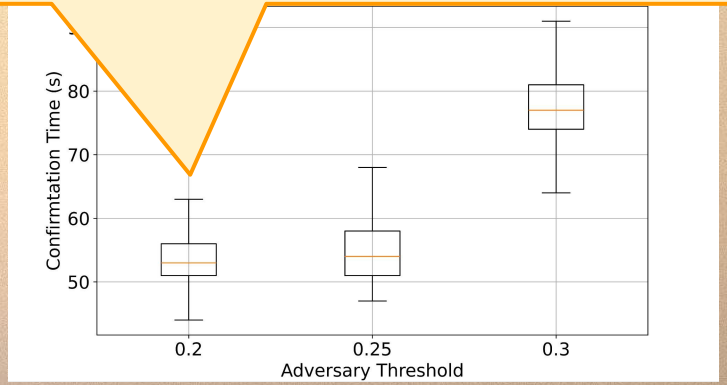
- Run the Confirmation Rule over 6 days
- 56 blocks were detected as re-orged, including a 2-block re-org.
- Confirmation Rule executed by polling a Beacon Node at 10s intervals, rather than executing it at the beginning of each slot
- FFG weight estimated from LDM weight

Results

- **None of the re-orged blocks were confirmed under the rule, no matter how the adversarial thresholds were tuned.**

If implemented directly into a beacon node,

we expect ≅ 12 sec



Paper

Our technical report is also linked to the talk description.

If you are interested in implementing this Fast Confirmation Rule, please get in touch with us!

You can find our contact details on the talk page.

Consensus Spec
PR #3339

Thank you!

DC7 SEA