THE GHOST-DAG PROTOCOL

Yonatan Sompolinsky[1] & Aviv Zohar[2]

The Hebrew University[1,2], DAGlabs[1], QED-it[2]

Bitcoin’s consensus protocol:

Behavior of honest miners:

1. Mine blocks that point to tip of longest chain
2. Publish blocks immediately

Security assumptions:

1. >50% of hashrate honest.
2. Block prop. << time between blocks

Then:

1. Pick the longest chain
2. The transaction set does not change (w.h.p. for txs with many confirmations)

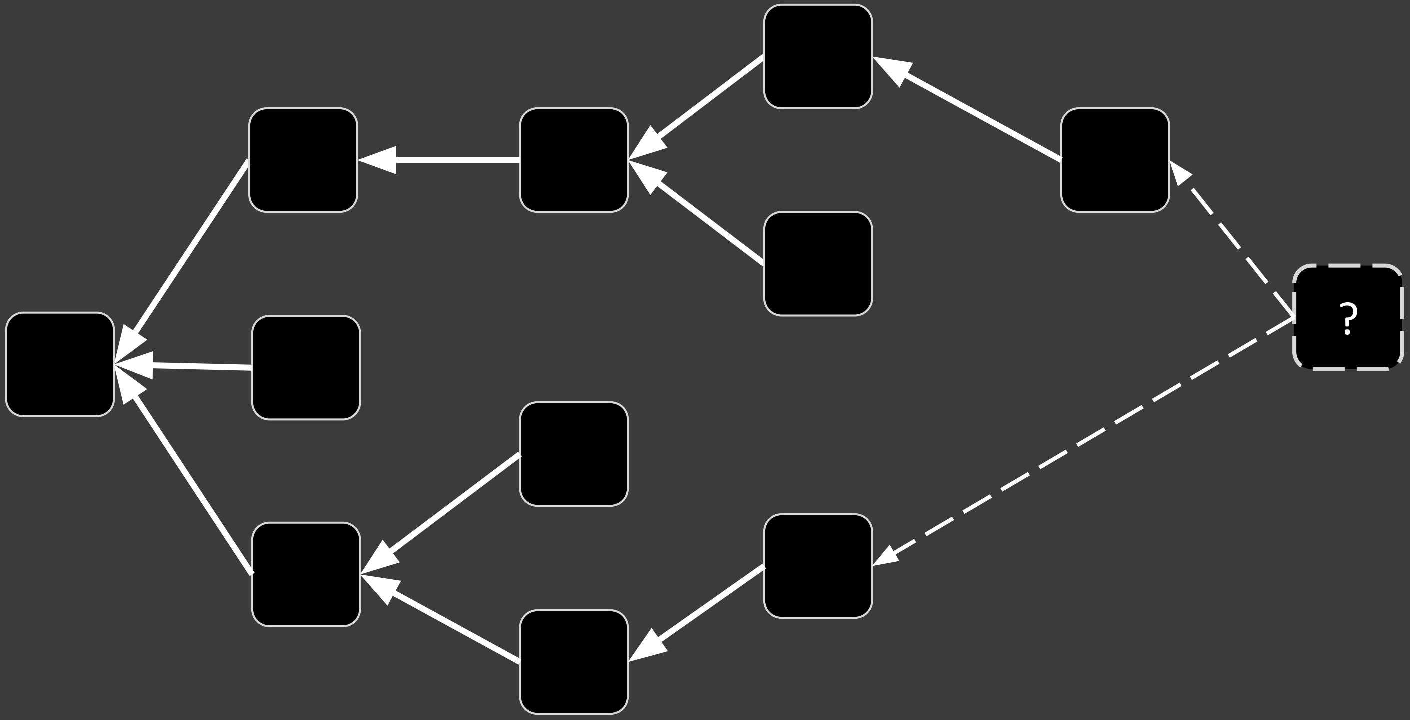
What you get

- Security no longer breaks at higher throughput

BUT NO FREE LUNCH!

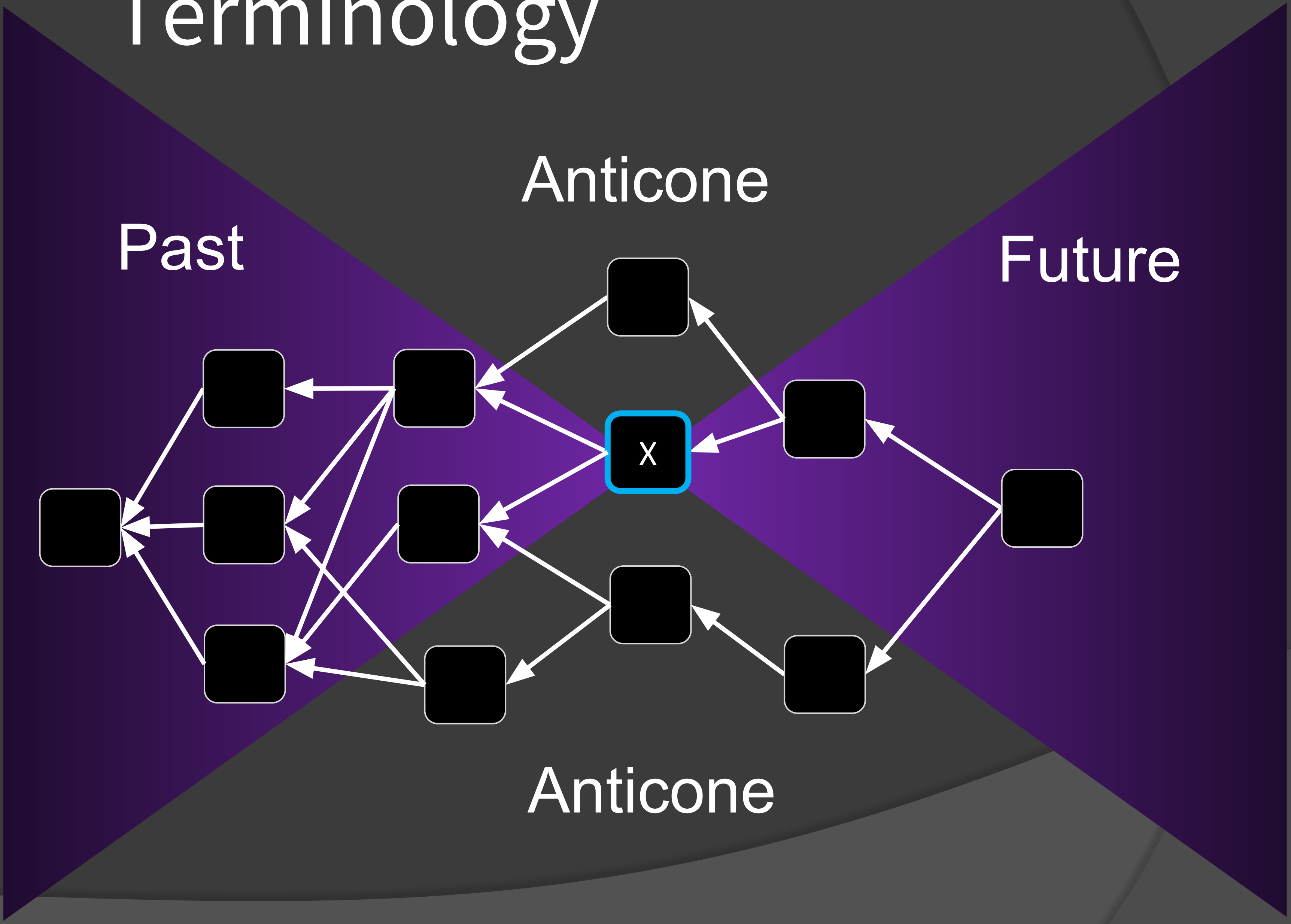
- Latency increases (Time until transactions are irreversible).
- (We also don’t solve other scaling issues like storage, validation time, bootstrapping, etc.)

From Chain to DAG



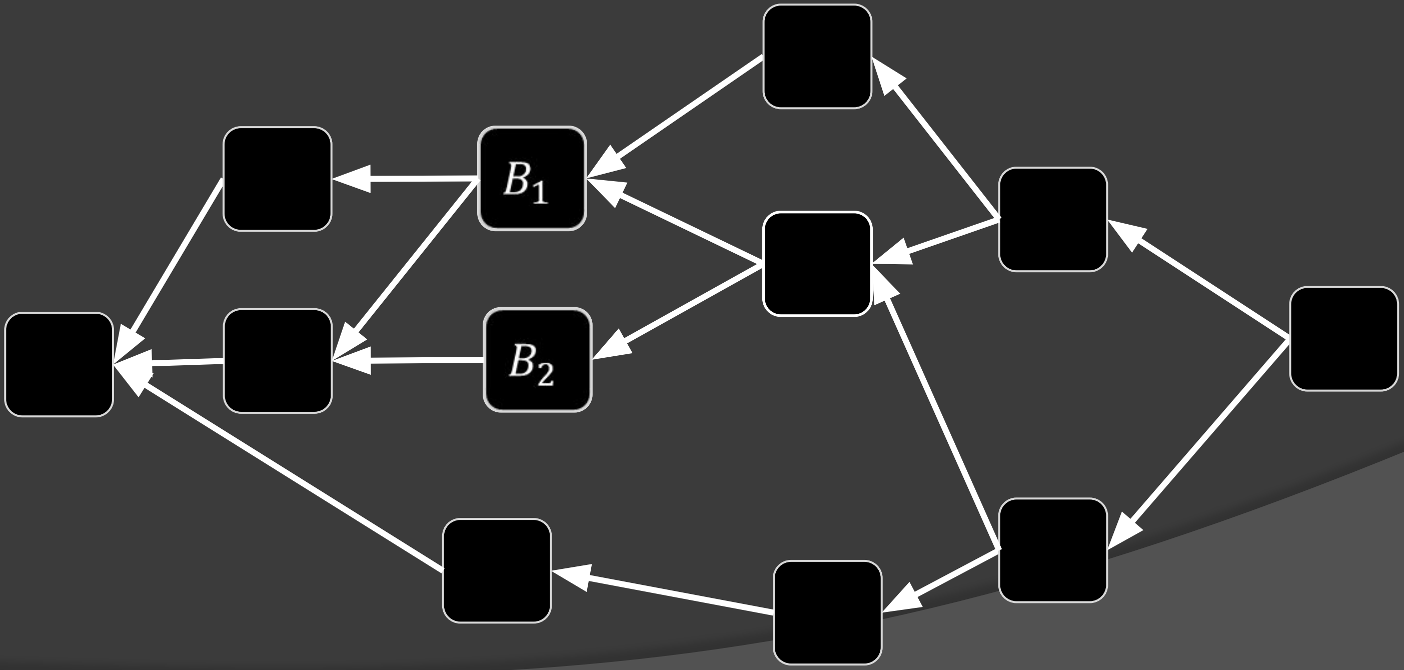
Goal: To get an ordering of the blocks that does not change

Terminology



What do "honest" blocks look like?

- Suppose blocks B_1, B_2 are "honest"
  - $B_1 \in Anticone(B_2)$ only if created roughly at the same time



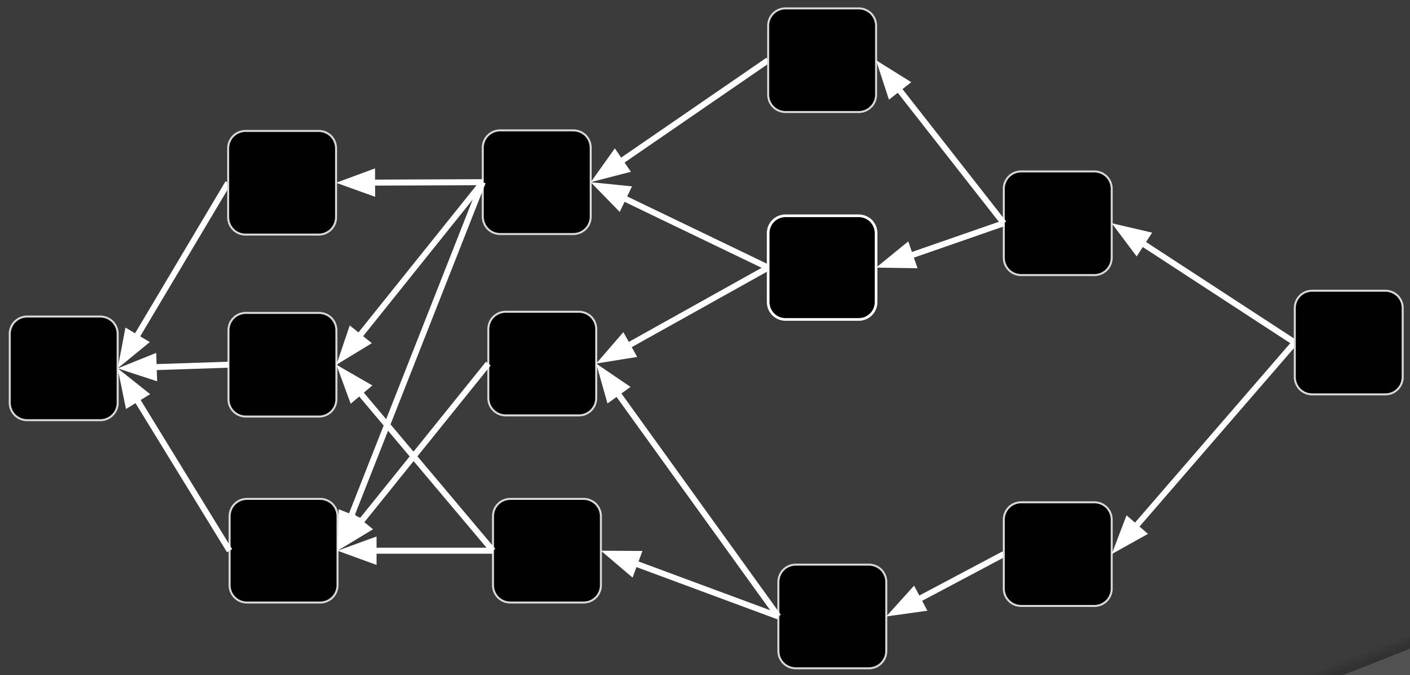
k-cluster (sometimes also called k-chain)

-

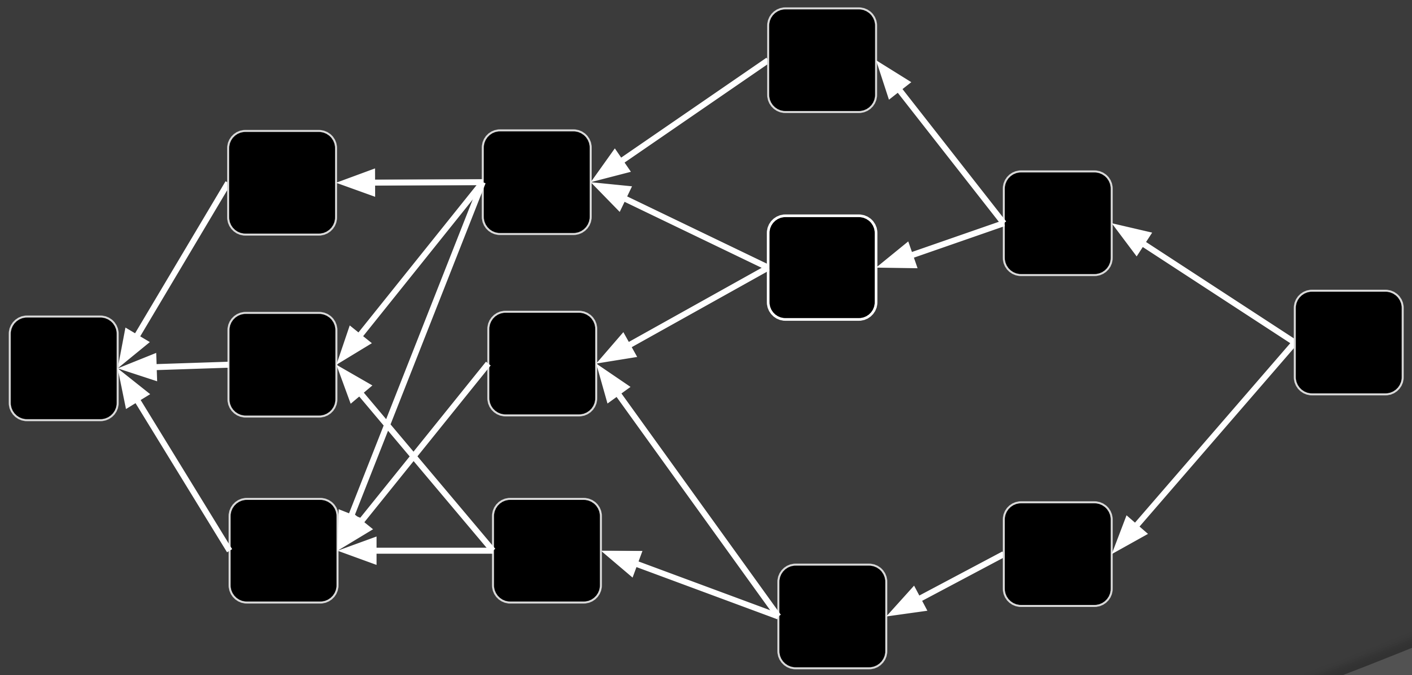
- A set of blocks C such that each block $B \in C$ has

$$|Anticone(B) \cap C| \leq \mathrm{k}$$

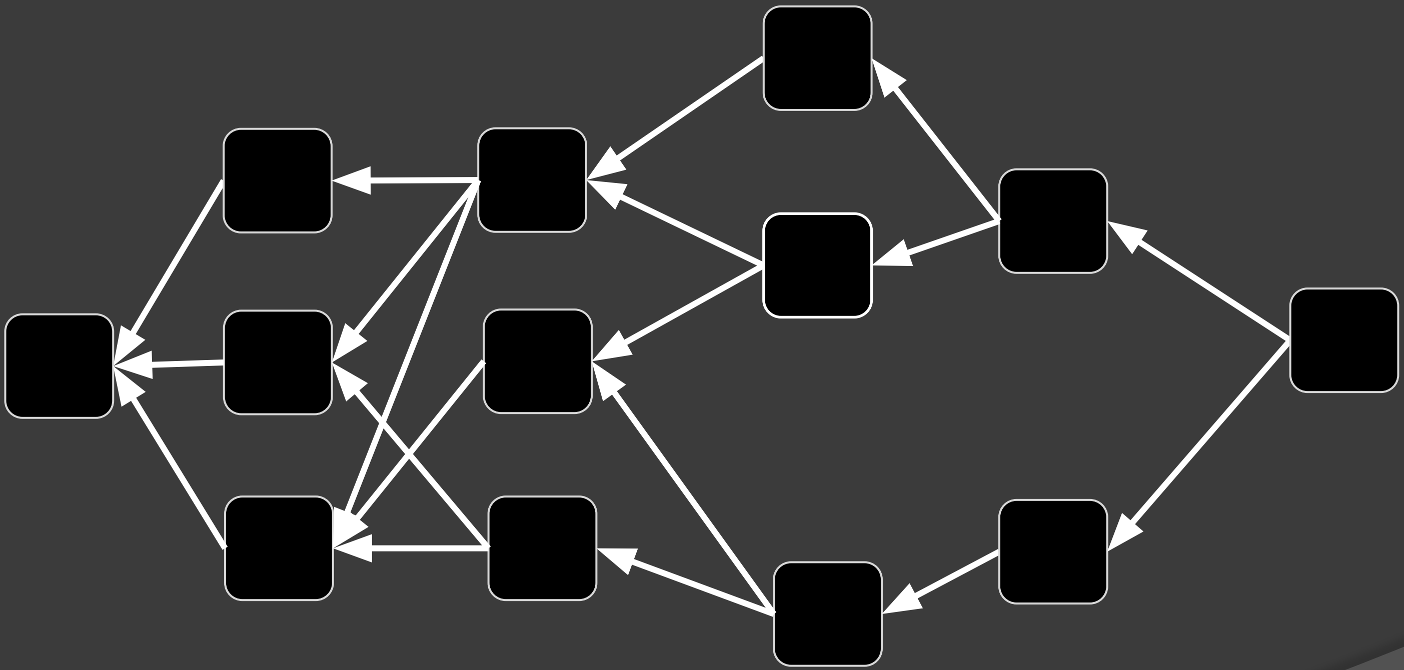
Example



K-Cluster



0-Cluster



The PHANTOM protocol:

1) Pick a max weight k-cluster in the DAG.

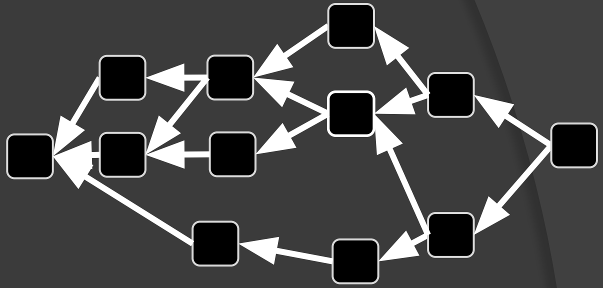
2) Sort it topologically in some canonical way*

*(that is past-dependent only).

It will contain most honest blocks
(if k was set well)

It will remain the same, thus the topological sort will not change (w.h.p).

Problem: It is generally hard (NP-Hard) to find the maximal k-cluster in a DAG.

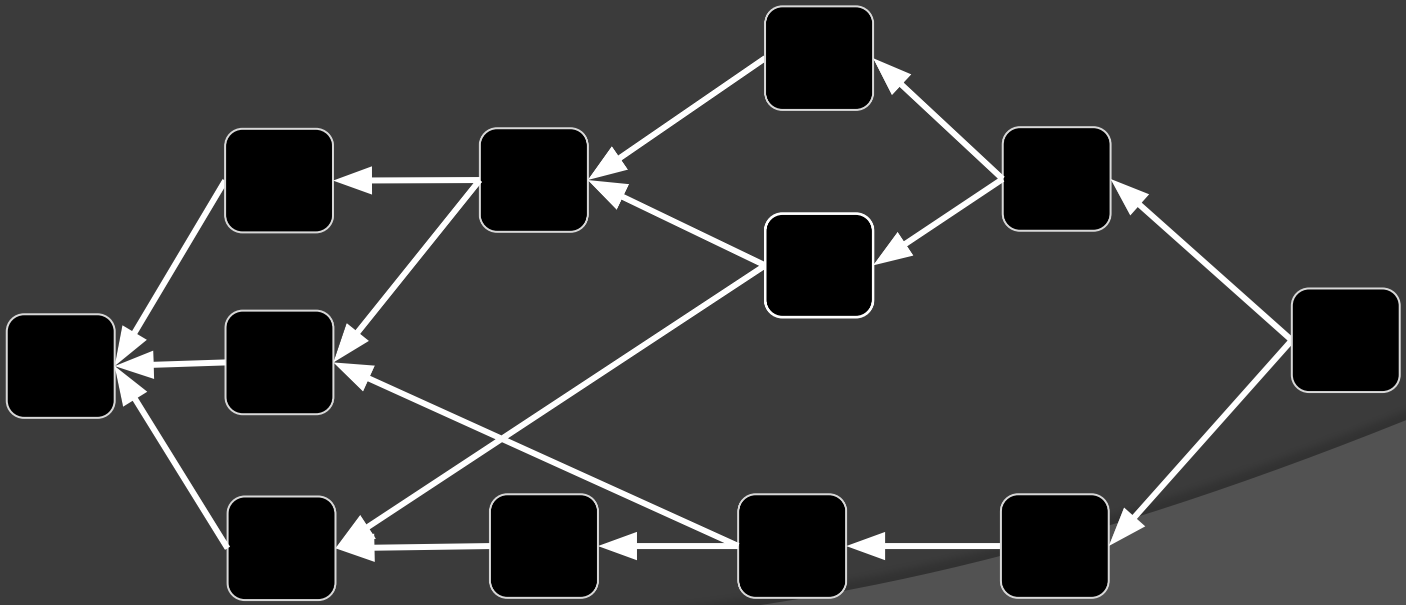


Solution: use a greedy algorithm to get a k-cluster

GHOST-DAG protocol

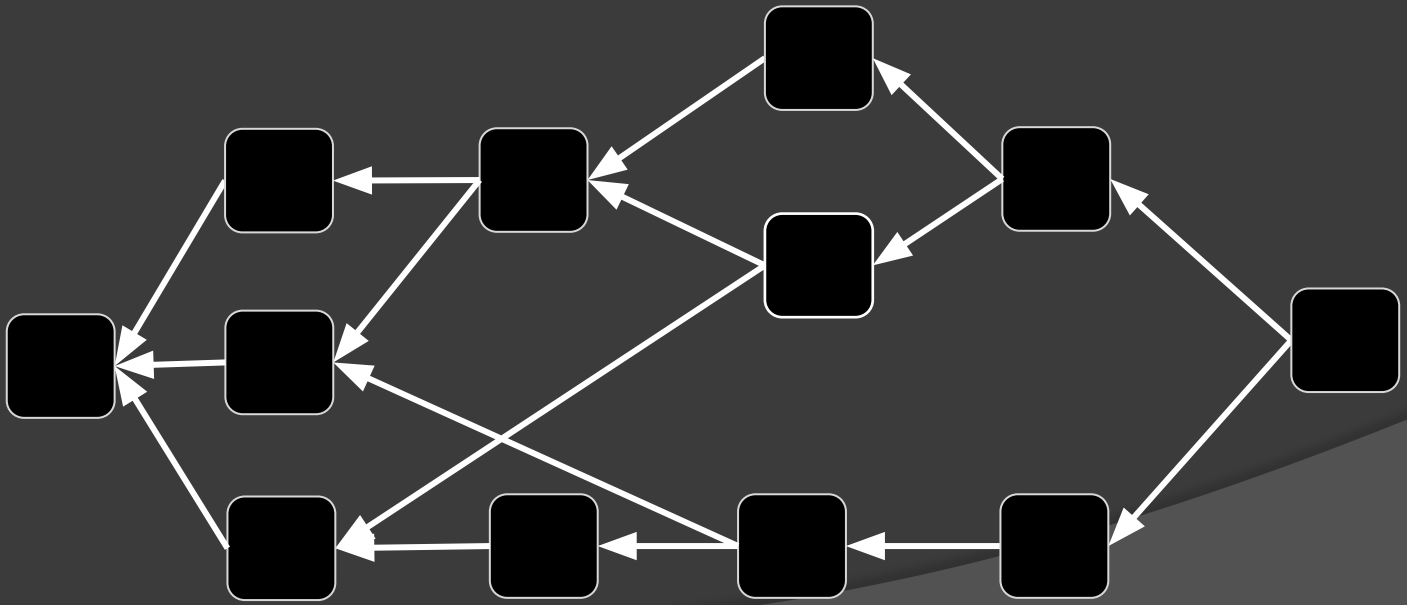
GHOST-DAG protocol

- Each block inherits the "heaviest" k-cluster from one of its predecessors.
- Adds blocks greedily (as long as still a k-cluster)



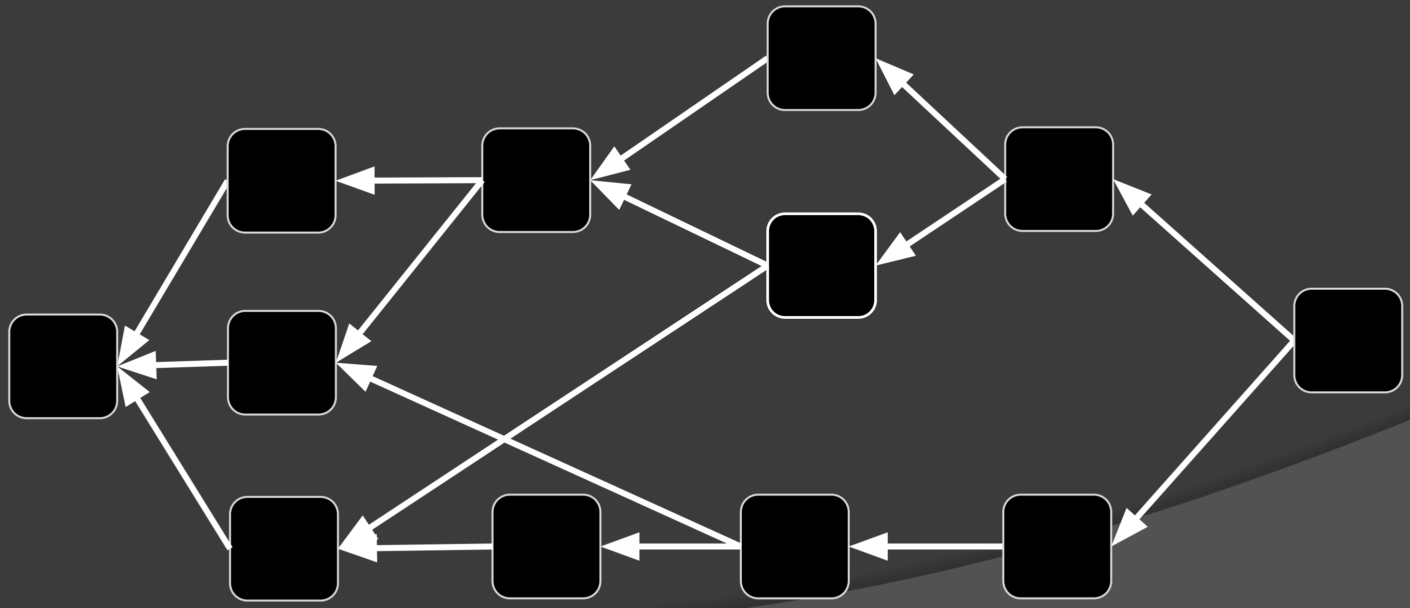
GHOST-DAG protocol

- Each block inherits the “heaviest” k-cluster from one of its predecessors.
- Adds blocks greedily (as long as still a k-cluster)



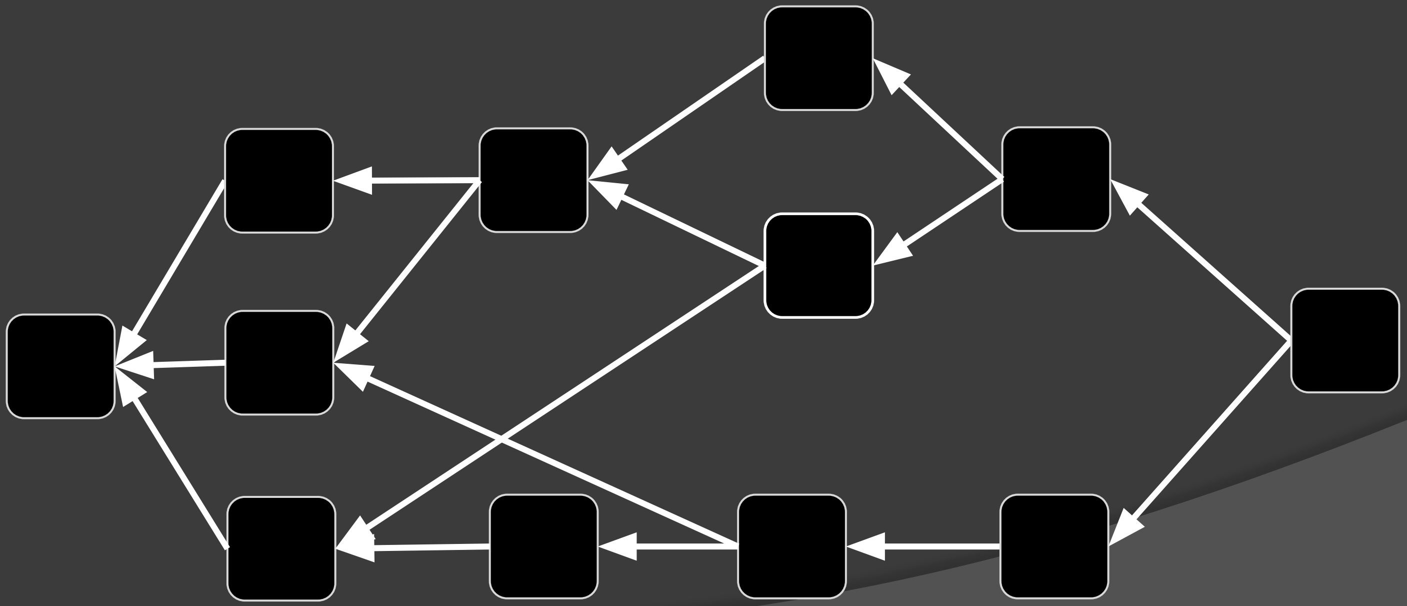
GHOST-DAG protocol

- Each block inherits the “heaviest” k-cluster from on of its predecessors.
- Adds blocks greedily (as long as still a k-cluster)



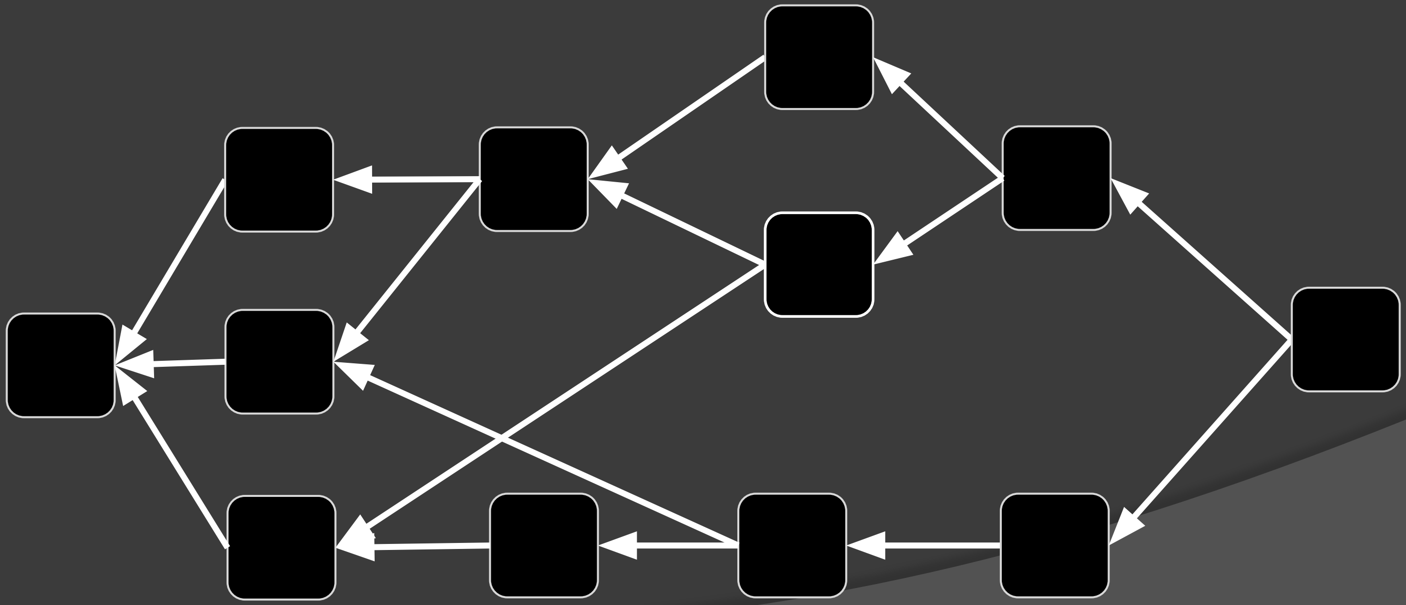
GHOST-DAG protocol

- Each block inherits the “heaviest” k-cluster from on of its predecessors.
- Adds blocks greedily (as long as still a k-cluster)

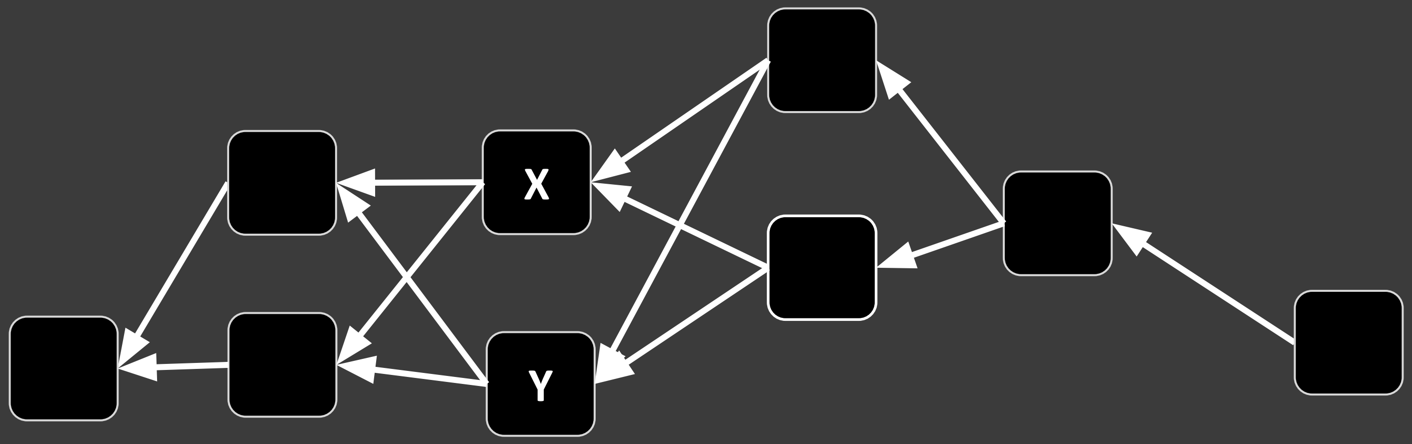


GHOST-DAG protocol

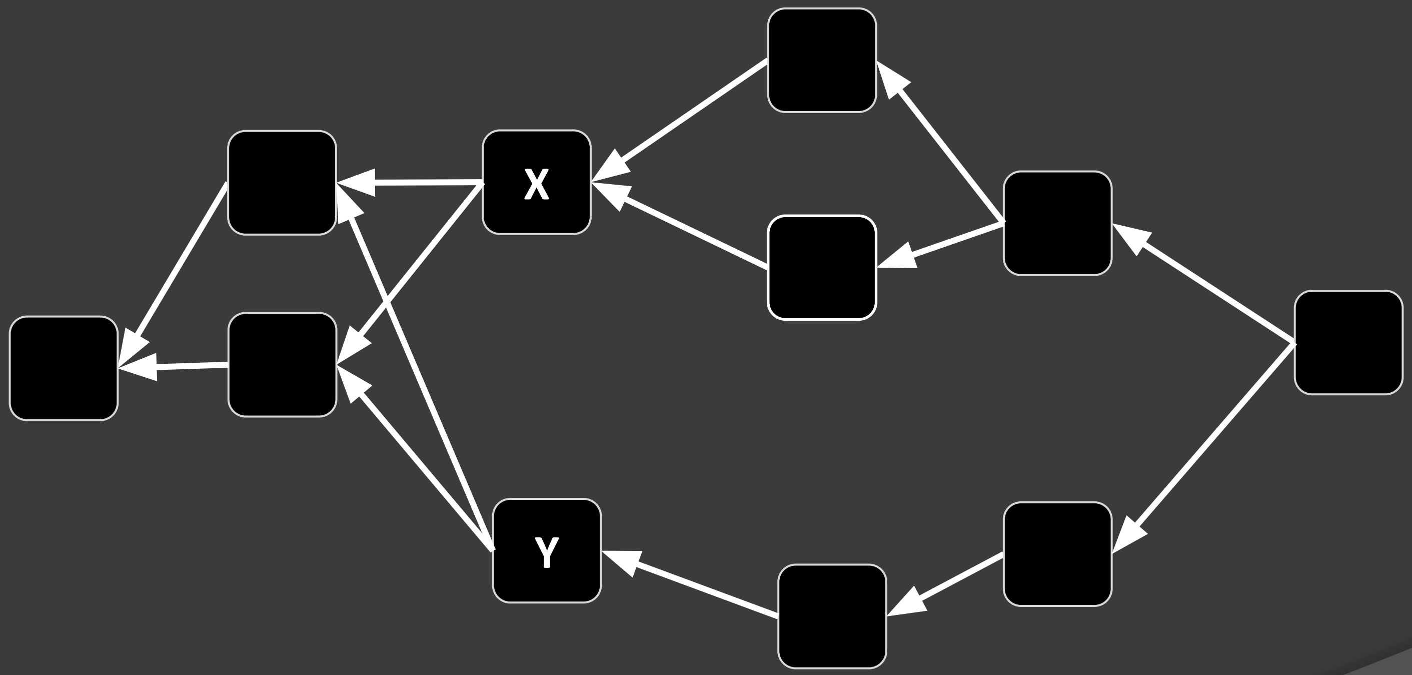
- Each block inherits the “heaviest” k-cluster from on of its predecessors.
- Adds blocks greedily (as long as still a k-cluster)



Intuition for resilience to double spends



Intuition for resilience to double spends



Comments

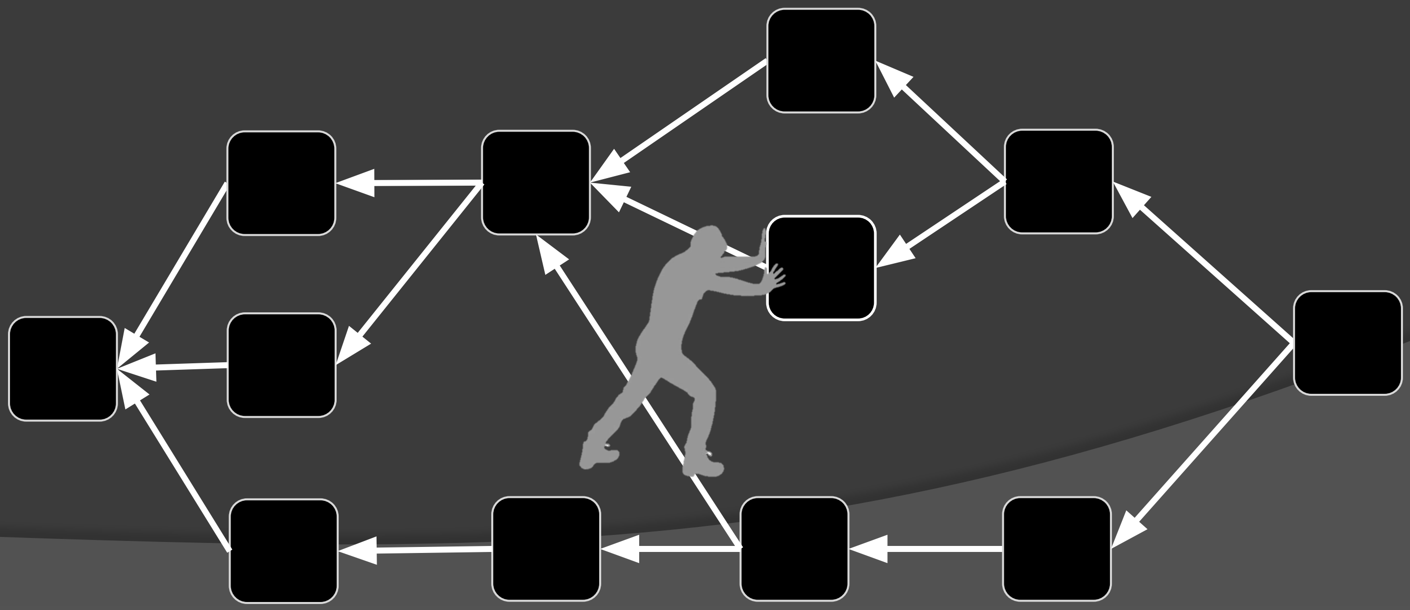
- Extra data-structures allow for very efficient implementation

- Topological order can also be “inherited” from “heaviest” predecessor

- All k-clutser blocks get block reward: Extra resilience to selfish mining

- Selfish mining:
  "push" honest blocks off the chain by strategically delaying block publication.

- To push blocks off largest k-cluster requires longer delays
  - Attacker more likely to lose block races



Thank you!

PHANTOM & GHOSTDAG full paper:
https://eprint.iacr.org/2018/104.pdf

Email:
avivz@cs.huji.ac.il
yoni_sompo@cs.huji.ac.il