Dfinity Consensus

Adam Carriker
Jannis Stoeter

Dfinity Overview

Dfinity Overview

- Proposed in 2018
    - Original Paper - Timo Hanke, Mahnush Movahedi and Dominic Williams
    - goal: “block times of a few seconds and transaction finality of only 2 confirmation”
- Dfinity Consensus
    - Analysis Paper - Ittai Abraham, Dahlia Malkhi, Kartik Nayak, and Ling Ren

Protocol

Dfinity Latency and Communication Complexity

Latency and Communication Complexity

- Types of adversaries

Latency and Communication Complexity

- Types of adversaries
  - Adaptive
  - Mildly / delayed adaptive
  - Static

Latency and Communication Complexity

- Types of adversaries
  - Adaptive
    - strongly adaptive, rushing/non-rushing, etc.
  - Mildly / delayed adaptive
    - must wait Δ time to corrupt party
  - Static
    - picks parties to corrupt before protocol starts

Latency and Communication Complexity

- We consider 2 types
  - Adaptive
  - Static

Latency and Communication Complexity

- We consider 2 types
  - Adaptive
    - can pick up to f parties to corrupt at any point
  - Static
    - picks up to f parties to corrupt before protocol starts

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
    - Worst case (think of adaptive adversary)

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case (think of adaptive adversary): $O(f*\Delta)$

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case (actual communication delay is $<< \Delta$)
    - pessimist case (actual communication delay is $= \Delta$)

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case (actual communication delay c << Δ)
      - only broadcast (step 1) must wait for 2Δ
      - all other communication happens at “network speed” (<< Δ)

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case (actual communication delay $c \ll \Delta$)
      - only broadcast (step 1) must wait for 2Δ
      - all other communication happens at “network speed” ($\ll \Delta$)
      - expected iterations until honest leader: 2
      - Invariant I + Invariant III: 2 iterations after honest leader will commit that leader’s proposed block

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case (actual communication delay $c \ll \Delta$)
      - only broadcast (step 1) must wait for 2Δ
      - all other communication happens at “network speed” ($\ll \Delta$)
      - expected iterations until honest leader: 2
      - Invariant I + Invariant III: 2 iterations after honest leader will commit that leader’s proposed block
      - 3 iterations * $(2\Delta) + 2\Delta = 8\Delta$

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c \ll \Delta$): 8Δ
    - pessimistic case (communication delay $c = \Delta$)

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: O(f*Δ)
  - Expected latency for block to be committed
    - optimistic case (c << Δ): 8Δ
    - pessimistic case (communication delay c = Δ)
      - assume f lowest-rank parties are Byzantine, certificate formed in (f+1)Δ
        - expected time for certificate to be formed is 2Δ

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case (communication delay $c = \Delta$)
      - assume f lowest-rank parties are Byzantine, certificate formed in $(f+1)\Delta$
        - expected time for certificate to be formed is 2Δ
      - expected iterations until honest leader: 2
      - Invariant I + Invariant III: 2 iteration after honest leader will commit that leader’s proposed block

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case (communication delay $c = \Delta$)
      - assume f lowest-rank parties are Byzantine, certificate formed in $(f+1)\Delta$
        - expected time for certificate to be formed is 2Δ
      - expected iterations until honest leader: 2
      - Invariant I + Invariant III: 2 iteration after honest leader will commit that leader's proposed block
      - 3 iterations * $(2\Delta + 2\Delta) + 2\Delta = 14\Delta$

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c \ll \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c \ll \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded
  - Fix: add equivocation check

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded
  - Fix: add equivocation check
    - Expected communication complexity
      - honest leader expected every 2 rounds
      - all honest parties send blocks to one another: $O(n^2)$

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded
  - Fix: add equivocation check
    - Expected communication: $O(n^2)$
    - Worst case: adaptive adversary

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded
  - Fix: add equivocation check
    - Expected communication: $O(n^2)$
    - Worst case: adaptive adversary
      - expected $O(f)$ iterations with Byzantine leader, so complexity is $O(n^3)$

Latency and Communication Complexity

- Types of adversaries - adaptive / static
- Latency
  - Worst case: $O(f^*\Delta)$
  - Expected latency for block to be committed
    - optimistic case ($c << \Delta$): 8Δ
    - pessimistic case ($c = \Delta$): 14Δ
- Communication complexity
  - Originally, unbounded
  - Fix: add equivocation check
    - Expected communication: $O(n^2)$
    - Worst case: $O(n^3)$

Relating Dfinity to Other Protocols

Dfinity vs O(1) Protocol

- What happens when we remove invariant II?

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]
- Latency (static / adaptive)

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]
- Latency (static / adaptive)
    - O(1) protocol has expected number of “rounds” = 13 (static) / 15 (adaptive)

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]
- Latency (static / adaptive)
  - O(1) protocol has expected number of “rounds” = 13 (static) / 15 (adaptive)
    - 1 “round” in O(1) = 2Δ in Dfinity

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs $[O(n^3)$ to $O(n^2)]$
- Latency (static / adaptive)
  - O(1) protocol has expected number of "rounds" = 13 (static) / 15 (adaptive)
    - 1 "round" in O(1) = 2Δ in Dfinity
  - O(1) static latency: 26Δ
  - O(1) adaptive latency: 30Δ

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]
- Latency (static / adaptive)
  - O(1) protocol has expected number of “rounds” = 13 (static) / 15 (adaptive)
    - 1 “round” in O(1) = 2Δ in Dfinity
  - O(1) static latency: 26Δ
  - O(1) adaptive latency: 30Δ
  - Dfinity: $8\Delta/14\Delta$

Dfinity vs O(1) Protocol

- What happens when we remove invariant II? O(1) protocol
- Number of Byzantine parties: same as Dfinity
- Communication complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]
- Latency (static / adaptive): 26Δ / 30Δ vs 8Δ / 14Δ

Dfinity vs Nakamoto Consensus

- Number of Byzantine

Dfinity vs Nakamoto Consensus

- Number of Byzantine: Same
- Number of blocks committed per round/iteration
    - Nakamoto:

Dfinity vs Nakamoto Consensus

- Number of Byzantine: Same
- Number of blocks committed per round/iteration
  - Nakamoto: 1
  - Dfinity:

Dfinity vs Nakamoto Consensus

- Number of Byzantine: Same
- Number of blocks committed per round/iteration
  - Nakamoto: 1
  - Dfinity: 0 or 1 or multiple
- Communication Complexity

Dfinity vs Nakamoto Consensus

- Number of Byzantine: Same
- Number of blocks committed per round/iteration
  - Nakamoto: 1
  - Dfinity: 0 or 1 or multiple
- Communication Complexity: $O(n^2)$ vs [$O(n^3)$ to $O(n^2)$]

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - With Nakamoto, Δ=10 seconds gives a block mine rate of 10 minutes
    - Transactions are considered committed after 6 blocks
    - 6 blocks * 10 minutes = 60 minutes to confirm a transaction

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - With Nakamoto, Δ=10 seconds gives a block mine rate of 10 minutes
    - Transactions are considered committed after 6 blocks
    - 6 blocks * 10 minutes = 60 minutes to confirm a transaction
  - With Dfinity, Δ=10 seconds and expected latency of 8Δ to 14Δ
    - [8*10s to 14*10s] = 80-140 seconds to confirm a transaction

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - With Nakamoto, Δ=10 seconds gives a block mine rate of 10 minutes
    - Transactions are considered committed after 6 blocks
    - 6 blocks * 10 minutes = 60 minutes to confirm a transaction
  - With Dfinity, Δ=10 seconds and expected latency of 8Δ to 14Δ
    - [8*10s to 14*10s] = 80-140 seconds to confirm a transaction
    - Recall Dfinity goal: “block times of a few seconds and transaction finality of only 2 confirmation”

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack?

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack? Yes

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack? Yes
  - Can Dfinity have Private Mining attack?

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack? Yes
  - Can Dfinity have Private Mining attack?
    - block verification procedure

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack? Yes
  - Can Dfinity have Private Mining attack?
    - block verification procedure
      - requires f+1 votes to certify a block
    - synchrony

Did Dfinity achieve its goal to be quicker & safer than Bitcoin?

- Latency (real-world)
  - Nakamoto: 60 minutes to confirm transaction
  - Dfinity: 80-140 seconds to confirm transaction
- Finality
  - Can Nakamoto consensus have Private Mining attack? Yes
  - Can Dfinity have Private Mining attack? No

Thinking Further

- What are some shortcomings you see with Dfinity?
- What are some possible improvements?