UNIVERSITÄT BERN

Consensus in blockchains: Theory and practice

Christian Cachin
University of Bern

ApPLIED, June 2024

BFTW^3: Why? When? Where? Workshop on Theory and Practice of Byzantine Fault Tolerance

Affiliated with DISC 2009

September 22, 2009
Elche, Spain

The workshop gathers researchers from both theory and systems communities and aims at understanding why the impressive research activity in the area of Byzantine fault-tolerance is not yet instantiated in practice. Has the moment for a wide deployment of BFT systems arrived, and if so, where BFT systems should be deployed in the first place?

Format

The workshop will consist of invited contributions. No published proceedings, the presentations may contain results that appeared or are going to appear elsewhere, work-in-progress reports, surveys and tutorials. A submission is expected to be a short (around two pages) abstract of the presentation.

u^b

UNIVERSITÄT BERN

BFTW^3: Why? When? Where?
Workshop on Theory and Practice of Byzantine Fault Tolerance

Affiliated with DISC 2009

September 22, 2009
Elche, Spain

However, there are few visible instantiations of these results in practical systems. Industrial software tends to ignore the BFT-related research and heads for less consistent but (apparently) simpler and more efficient solutions (e.g., [5, 16, 11]).

In this workshop, we discussed the state of the art in BFT systems, and tried to understand why BFT systems have not seen a widespread adoption, and what we could do to increase the chances of deploying BFT systems.

A history of consensus protocol development

- **1985 until 2000:** Theory research, many theorems, no systems, no prototypes
- **2000 until 2010:** Systems research, many prototypes, no products
- **2015– :** Deployment in practice with cryptocurrencies
- **Today:** More theory research, systems research, products, and deployments

Overview

- **for** *model* $\in$ nine kinds of blockchain consensus **do**
  - describe *model*

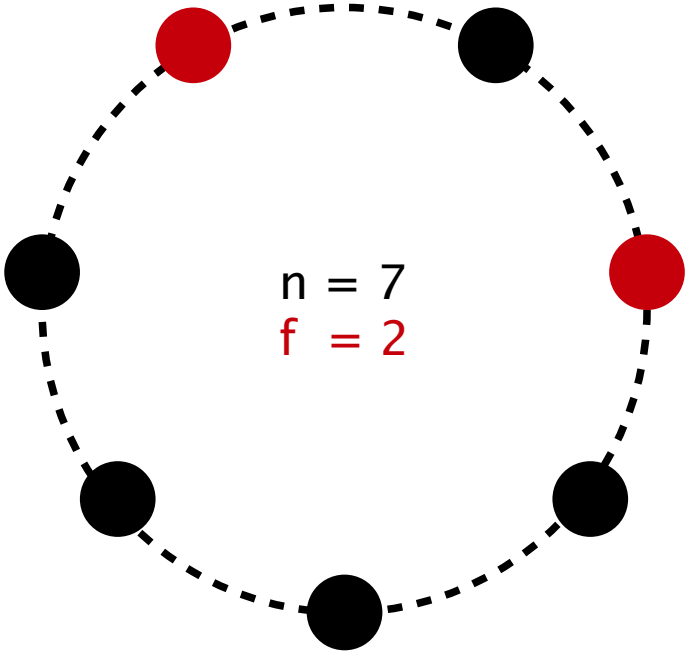
- **for** *model = probabilistic voting* **do**
  - theory of *Snow protocols*
  - practice of *Avalanche blockchain and AVAX cryptocurrency*

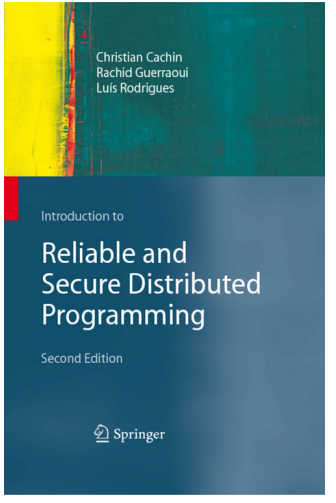
- Answer your questions

Consensus overview

1 – Threshold trust (BFT)

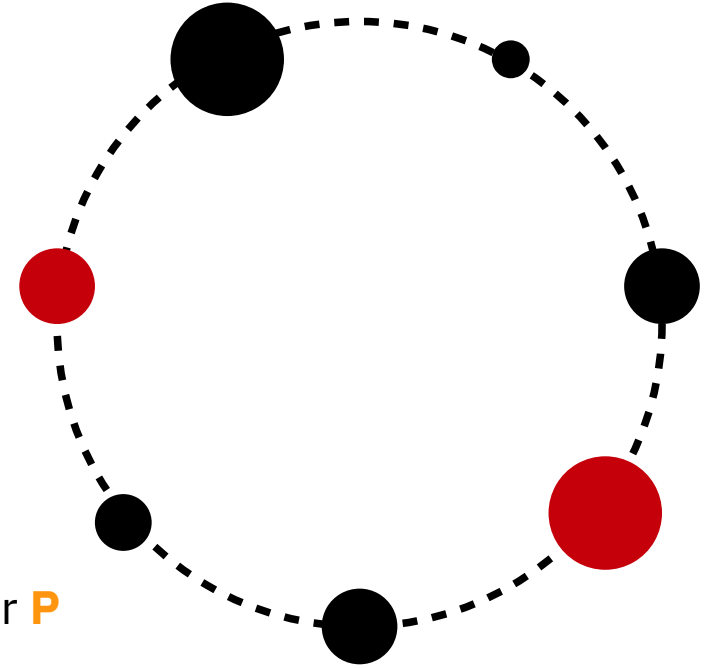
- Trust by numbers
  - n nodes total
  - f faulty (Byzantine) nodes
- Nodes are identified
  - Proof-of-Authority (PoA)
- Homogeneous and symmetric
- Requires n > 3f
- Tendermint/Cosmos, Internet Computer (DFINITY), VeChain, BNB SC, Hashgraph, TRON …





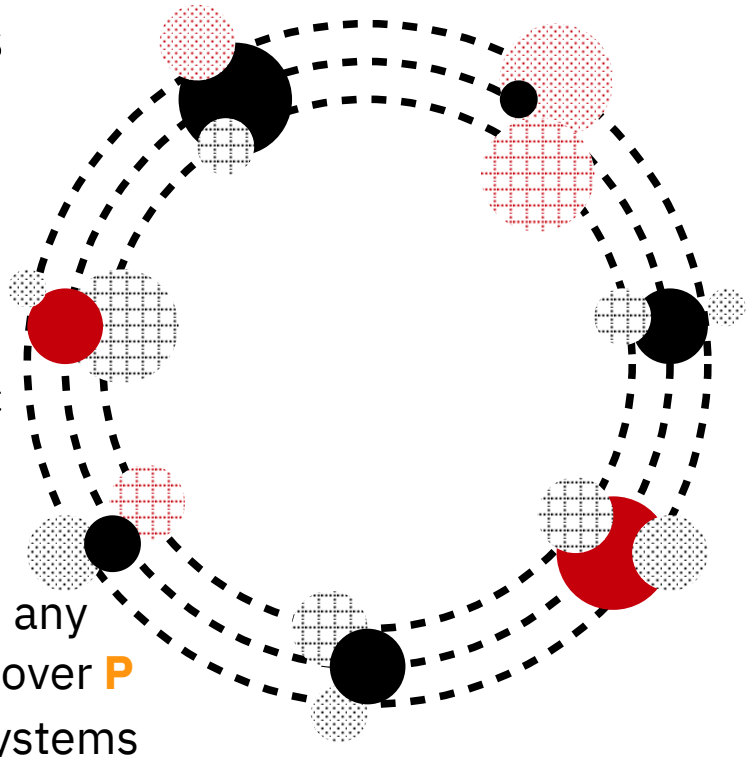
2 – Generalized trust

- Trust by generalized quorums
  - Set of nodes **P**
  - Fail-prone sets consisting of possibly Byzantine nodes
  - Byzantine quorum system
- Heterogeneous and symmetric
- Requires Q3-property
  - Any 3 fail-prone sets must not cover **P**
- Not used by any cryptocurrency (!)



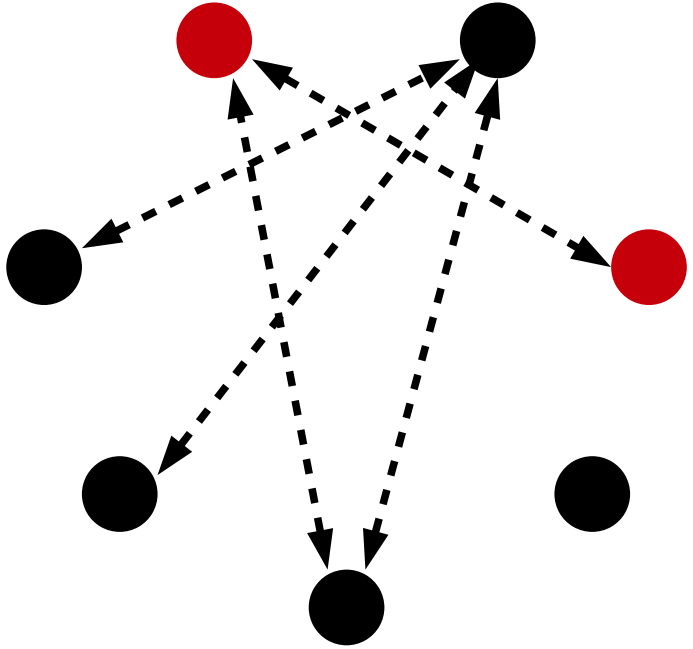
3 – Asymmetric trust

- Subjective generalized quorums
- Every node has its own Byz. quorum system on **P**
- Heterogeneous and asymmetric
- Requires B3-property
  - ∀ p, p' : any fail-prone set of p with any set of p' and any of both must not cover **P**
  - Consistent across nodes quorum systems
- Ripple, Stellar, [CT19]



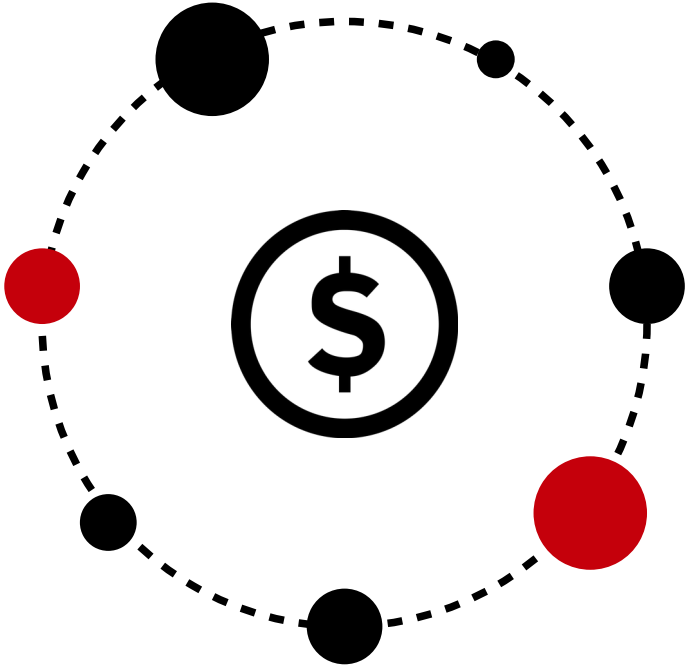
4 – Unstructured, probabilistic voting

- Random sampling of peers
- Exchange information and votes
- Often coupled with a DAG (directed acyclic graph) on transactions
- Avalanche, Conflux, IOTA-Tangle



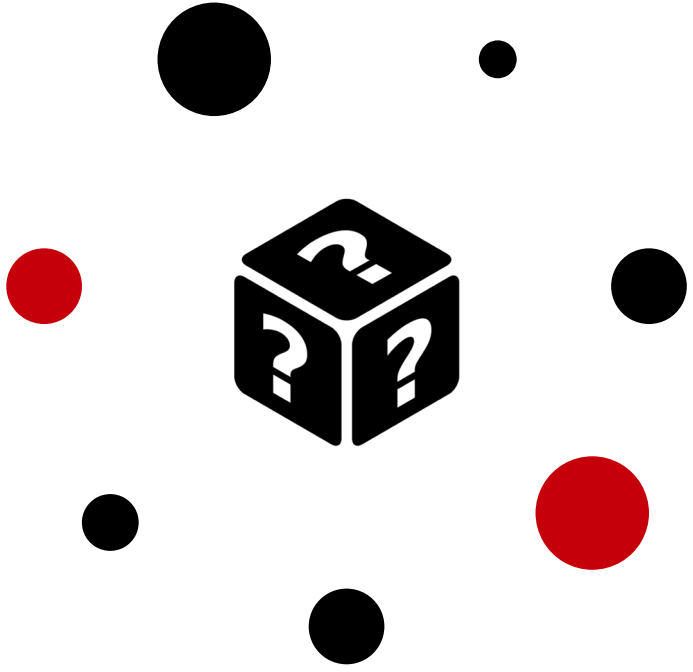
5 – Stake-based voting

- Stake determines voting power
  - Including delegated stake (DPoS)
- Protocols generalized from symmetric voting (BFT)
- Slashing of invested stake upon detection of misbehavior
- Tendermint/Cosmos, EOS, NEO, Aptos, SUI, BNB SC …



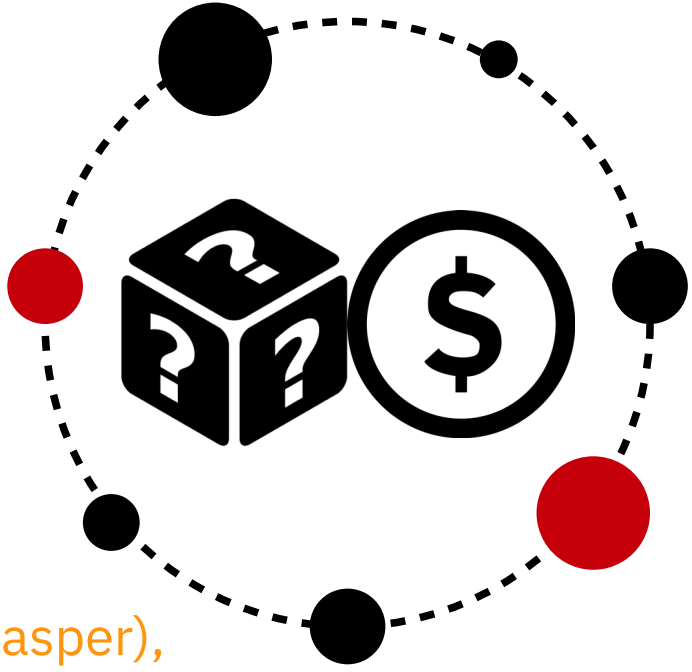
6 – Stake-based probabilistic choice

- Lottery according to stake
- Probabilistic leader election
- Cryptographic sortition using a verifiable random function (VRF)
- Cardano/Ouroboros …



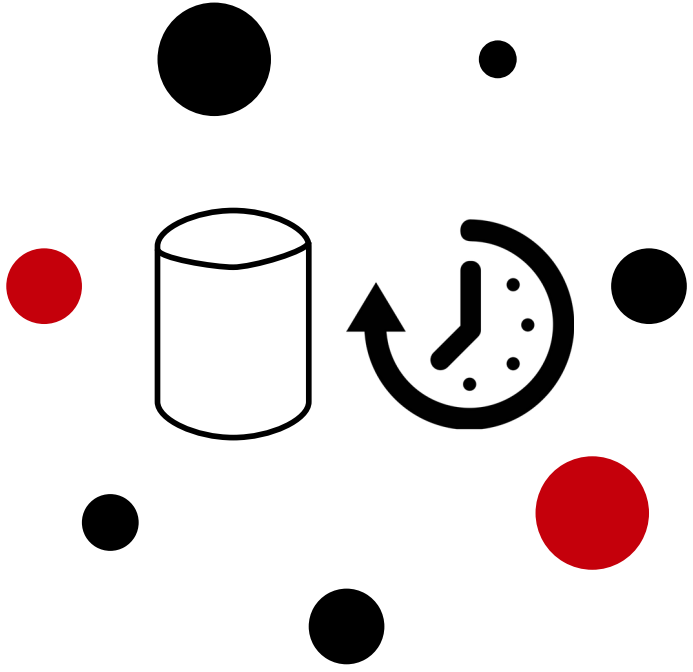
7 – Hybrid prob. choice and stake voting

- Stake determines probability or voting power
- Mix of random choice with voting
- Slashing of invested stake upon detection of misbehavior
- Ethereum (LMD-GHOST & FFG-Casper), Polkadot (BABE & GRANDPA), Algorand ...



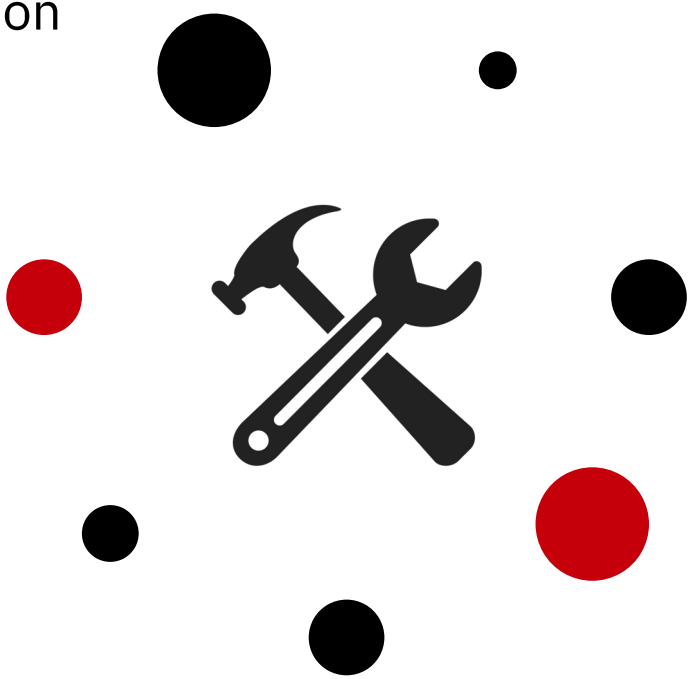
8 – Proof-of-space and proof-of-delay

- Storage space as resource
- Cryptographic ZK proofs for storage at particular time
- Time delay to prove storage investment over time
- Filecoin, Chia, Storj …



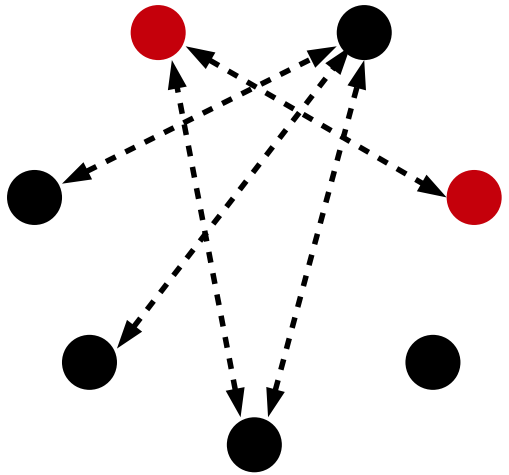
9 – Proof-of-work

- Demonstrate invested computation
- Nakamoto consensus
- Bitcoin and variations,
  Litecoin, Dogecoin,
  Ethereum (1.0) and variations,
  Ethereum Classic,
  Monero, ZCash ...



Snow and Avalanche consensus

Recent results with Ignacio Amores-Sesar &
Philipp Schneider & Enrico Tedeschi



Avalanche

- Avalanche is a prominent layer-1 blockchain
  - AVAX cryptocurrency
  - Smart-contract platform
  - AVAX is in the top 15 by market cap

- Novel approach to consensus
  - "Snow family" of protocols:
    Slush → Snowflake → Snowball → [Snowman →] Avalanche

  - Introduced in a white paper 2019:
    Scalable and Probabilistic Leaderless BFT Consensus through Metastability (Yin, Sekniqi, van Renesse, Sirer)

  - Based on random sampling of peer nodes

Recent results [ACT22, ACT24]

- [ACS24]
  - Analysis of the consensus dynamics
  - Proofs for safety and liveness of (idealized) Snow protocols
  - Binary consensus

- [ACT22]
  - Detailed pseudocode of DAG-structured ledger consensus protocol
  - First independent analysis
  - Illustrated some problems and provided a solution
  - Generic broadcast (not quite atomic broadcast)

Avalanche network model

UNIVERSITÄT BERN

- Cryptocurrency and smart-contract platform
  - X-chain: eXchange (AVAX currency, other tokens)
  - P-chain: Platform (validator node management, staking)
  - C-chain: smart Contracts (EVM-compatible), with application-specific subnets

- n validator nodes
  - Each validator stakes 2000 AVAX (≈ 60'000 USD, June 2024)
  - $n \approx 1500$ (June 2024)
  - Throughput: ≈ 10 tps (on average); 50-100 tps (max. recorded); 4500 tps (max. claimed)

- Security
  - Tolerates faulty (Byzantine) nodes
  - Secure "only" against corruption of up to $\sqrt{n}$ nodes

Problem statement

- Consensus is binary
  - All nodes *propose* 0 or 1
  - All correct nodes have *stabilized* on the same value – or – they *decide* the same value
- Protocol operates in synchronous rounds
  - Number of rounds T
  - Security parameter β
- Randomized protocol
  - Every node sends and receives O(k) messages per round
- Termination
  - All correct nodes terminate after T rounds, except with probability negligible in β

Goals

- Fix k as small constant
  - $O(n)$ messages overall

- Number of rounds T
  - should be logarithmic in n
  - should be polynomial in β

- Related to the literature on dynamics of consensus
  - Overview by Becchetti, Clementi, Natale (SIGACT News, 2020)

Slush consensus stabilization

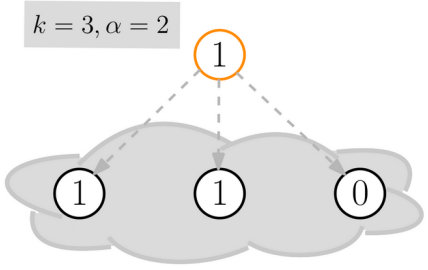
- $b \in \{0,1\}$ // consensus on a bit

- **for** round = 1, …, T **do**
  - pick k random parties, query them for their bit b
  - **if** at least α answers are b^* **then** // $\alpha > k/2$
    $b \leftarrow b^*$
- decide(b)

Slush consensus stabilization

- $b \in \{0,1\}$ // consensus on a bit

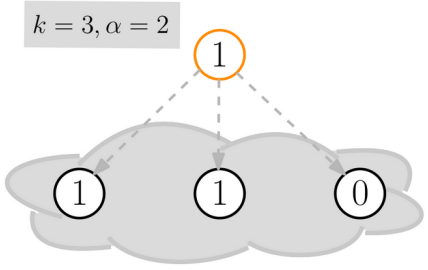
- **for** round = 1, ..., T **do**
  - pick k random parties, query them for their bit b
  - **if** at least α answers are b* **then** // $\alpha > k/2$
    $b \leftarrow b^*$
- decide(b)

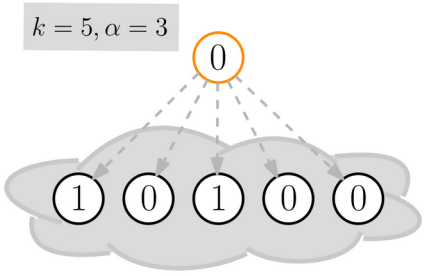


Slush consensus stabilization

- $b \in \{0,1\}$ // consensus on a bit

- **for** round = 1, ..., T **do**
  - pick k random parties, query them for their bit b
  - **if** at least α answers are b* **then** // $\alpha > k/2$
    $b \leftarrow b^*$
- decide(b)

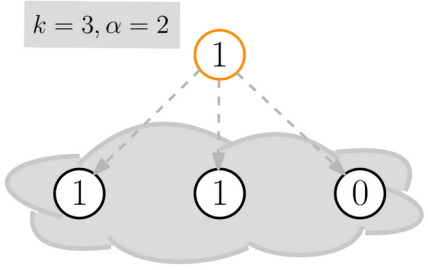


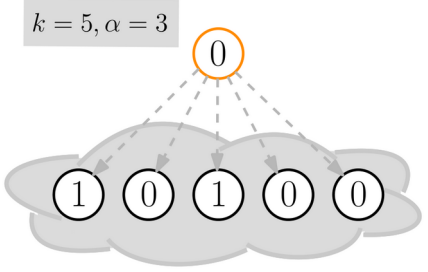


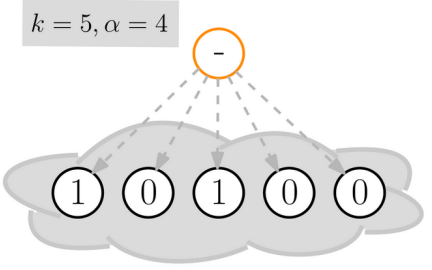
Slush consensus stabilization

- b ∈ {0,1} // consensus on a bit

- **for** round = 1, …, T **do**
  - pick k random parties, query them for their bit b
  - **if** at least α answers are b* **then** // α > k/2
    b ← b*
- decide(b)







How does Slush perform?

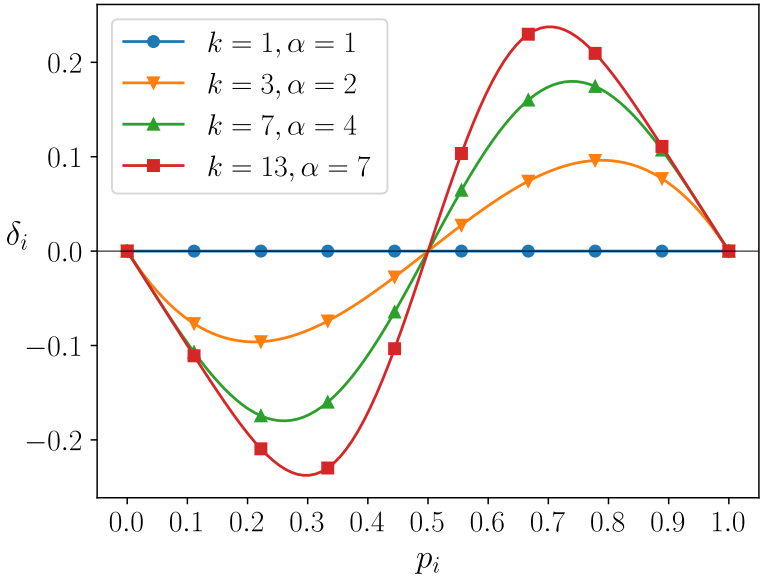
- Let p_i be fraction of nodes with opinion 1 in round i
- Let $\delta_i \in [-1, 1]$ be the expected "progress" towards consensus on 1
- For fixed k and α, the progress δ_i is a function of p_i :

$$\delta_i = \delta(p_i) := \sum_{\ell=\alpha}^{k} \binom{k}{\ell} \left[p_i^{\ell}(1-p_i)^{k-\ell+1} - (1-p_i)^{\ell} p_i^{k-\ell+1} \right].$$

How does Slush perform?

- Let p_i be fraction of nodes with opinion 1 in round i
- Let $\delta_i \in [-1, 1]$ be the expected "progress" towards consensus on 1
- For fixed k and α, the progress δ_i is a function of p_i :

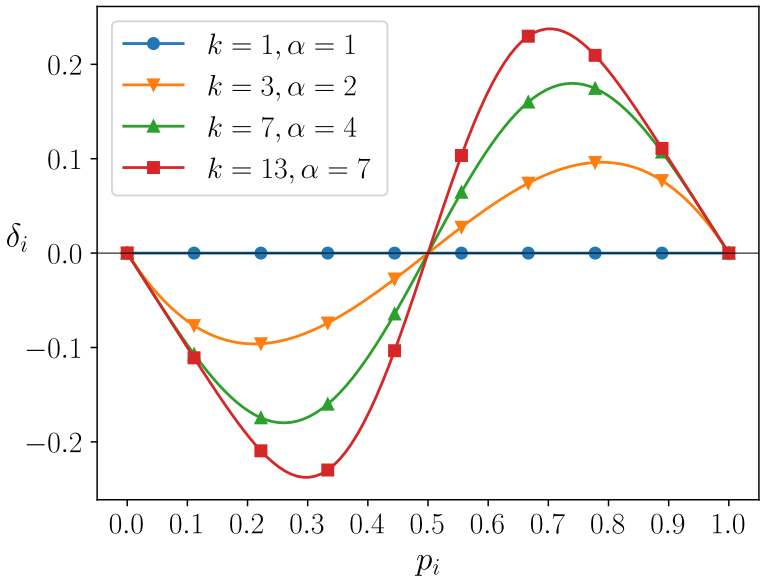
$$\delta_i = \delta(p_i) := \sum_{\ell=\alpha}^{k} \binom{k}{\ell} \left[p_i^{\ell}(1-p_i)^{k-\ell+1} - (1-p_i)^{\ell} p_i^{k-\ell+1} \right].$$

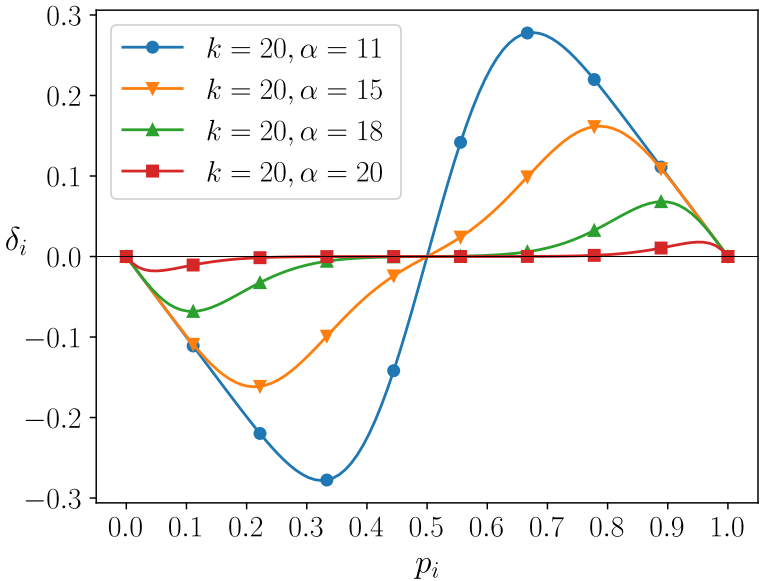


How does Slush perform?

- Let p_i be fraction of nodes with opinion 1 in round i
- Let $\delta_i \in [-1, 1]$ be the expected "progress" towards consensus on 1
- For fixed k and α, the progress δ_i is a function of p_i :

$$\delta_i = \delta(p_i) := \sum_{\ell=\alpha}^{k} \binom{k}{\ell} \left[p_i^{\ell}(1-p_i)^{k-\ell+1} - (1-p_i)^{\ell} p_i^{k-\ell+1} \right].$$





Results for Slush and consensus stabilization

- Theorem 1: For $k \geq 2$ and $\alpha = (k+1)/2$, Slush reaches stable consensus in

  $$O(\log n + \beta)$$

  rounds, with all but negligible probability in β and up to $O(\sqrt{n})$ corrupted nodes.

- Theorem 2: For $k \geq 2$ and $k/2 < \alpha < k$, the expected number of rounds for Slush rounds to reach a stable consensus is

  $$\Omega(\log n / \log k),$$

  with up to $O(\sqrt{n})$ corrupted nodes.

Consensus with a decision: Snowflake ...

UNIVERSITÄT BERN

- $b \in \{0,1\}$ // consensus on a bit
- counter ← 0
- **while** counter < β **do**
  - pick k random parties, query them for their bit b
  - **if** at least α answers are $b^* \neq b$ **then** // $\alpha > k/2$, Snowflake termination condition (+)
    - b ← b*
    - counter ← 0
  - **else**
    - counter ← counter + 1
- decide(b) // decide after β queries with a majority for b

- Snowball changes the termination condition (+)
  - **if** more rounds exist ever with ≥ α ans. for $b^* \neq b$ than rounds with ≥ α ans. for b **then ...**

Analysis of Snowflake and Snowball

- Theorem 3: In Snowflake and Snowball, with a (weak) adversary, these two properties are mutually exclusive:

  1) Consensus holds with all but negligible probability (in β);

  2) Correct parties decide after polynomially many (in β) rounds.

A better tradeoff for consensus: Blizzard

- Blizzard changes the termination condition (+) again
  - c_0 counts number of rounds ever with an α-majority for 0
  - c_1 counts number of rounds ever with an α-majority for 1

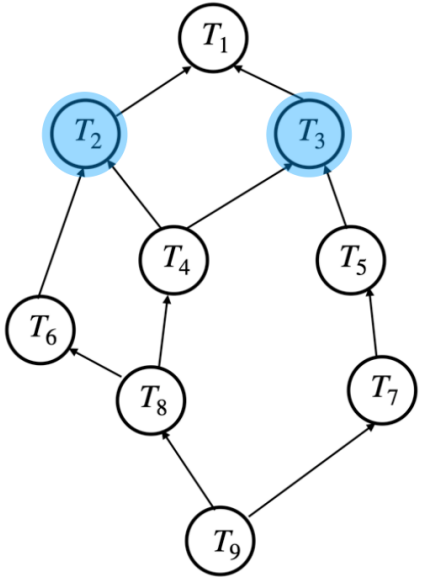
- New termination condition (+): stop when their difference exceeds some t
  - **if** $| c_0 - c_1 | \geq t$ **then** ...

- Theorem 4: Blizzard reaches consensus with all but negligible probability (in β) and terminates in up to $O(\log n + \beta)$ rounds.

DAG-ledger consensus (generic broadcast)

UNIVERSITÄT BERN

- Used in X-Chain
  - Extends consensus to a broadcast protocol
- Transactions form a DAG, a directed acyclic graph
- Transactions without dependencies (T2 and T3) may be delivered (accepted) in any order
  - "Generic broadcast" parameterized by a conflict relation (weaker than atomic broadcast)
- Transactions that conflict must be ordered



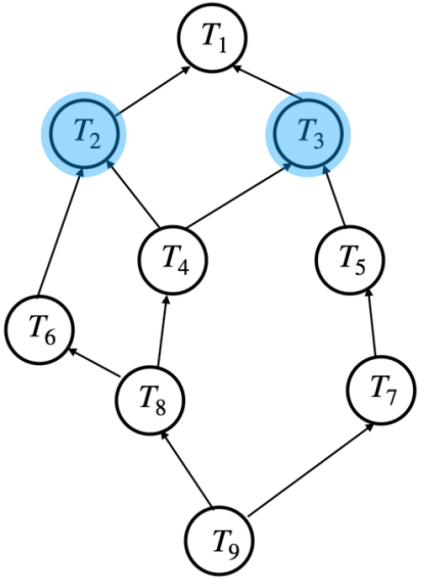
T2 and T2 independent

- In principle, every transaction is decided with a Snowball-like protocol

Avalanche DAG-ledger consensus

- **while** TRUE **do**
  - select some transaction T
  - pick k random parties and query them about T

  - **if** more than α positive results **then**
    update DAG: for every ancestor T' of T,
    increment counter(T') for acceptance
  - **else**
    update DAG: for every ancestor T' of T,
    reset (to 0) counter(T') for acceptance
  - **if** (∃ T* that is not conflicting ∧ counter(T*) ≥ β1) ∨
    (∃ T* that is conflicting ∧ counter(T*) ≥ β2) **then**
    output ("deliver") T



Conflicting tx can come to exist in the DAG.

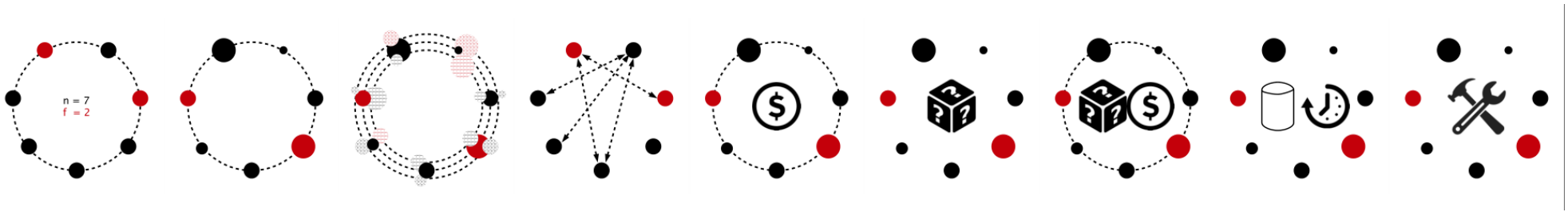
Referencing them cleverly can delay acceptance of innocent tx.

Analysis of DAG-ledger consensus

- Detailed pseudocode of Avalanche protocol
- Identified a liveness problem
  - Adversary may delay acceptance of a victim transaction arbitrarily
- For other reasons, Ava Labs/Avalanche abandons the DAG protocol on the X-chain in March '23

Conclusion

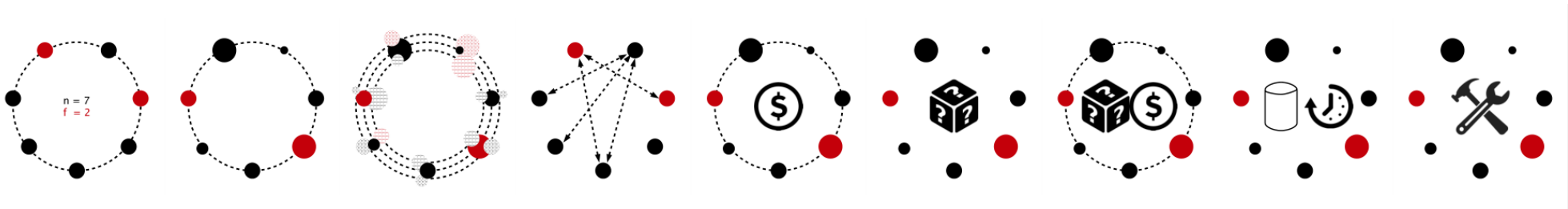
- Byzantine-tolerant consensus protocols matter and are here to stay
- Assumptions are more important than protocols



Conclusion

UNIVERSITÄT BERN

- Byzantine-tolerant consensus protocols matter and are here to stay
- Assumptions are more important than protocols



- Avalanche: Efficient probabilistic protocols, interesting consensus dynamics

- Links
  - Web: https://crypto.unibe.ch/
  - Blog: https://cryptobern.github.io/
  - Twitter/X: https://x.com/cczurich/

Thanks

- This work has been supported by
  - Swiss National Science Foundation (SNSF);
  - Donation from Avalanche, Inc.; and a
  - Sui Academic Research Award.

- Links
  - Web: https://crypto.unibe.ch/
  - Blog: https://cryptobern.github.io/
  - Twitter/X: https://x.com/cczurich/

References

Literature

- Model 1: Threshold trust
  - [CMSZ22] Cachin, C., Mićić, J., Steinhauer, N. & Zanolini, L. (2022). Quick Order Fairness. Proc. Financial Cryptography and Data Security (FC), LNCS 13411, 316–333. https://doi.org/10.1007/978-3-031-18283-9_15

Literature

UNIVERSITÄT BERN

- Model 2: Generalized trust
  - [AC20] Alpos, O., & Cachin, C. (2020). Consensus Beyond Thresholds: Generalized Byzantine Quorums Made Live. Proc. 39th Symposium on Reliable Distributed Systems (SRDS), 31–40. https://doi.org/10.1109/SRDS51746.2020.00010
  - [ACZ21] Alpos, O., Cachin, C., & Zanolini, L. (2021). How to Trust Strangers: Composition of Byzantine Quorum Systems. Proc. 40th Symposium on Reliable Distributed Systems (SRDS), 120–131. https://doi.org/10.1109/SRDS53918.2021.00021
  - [AC23] Alpos, O. & Cachin, C. (2023). Do Not Trust in Numbers: Practical Distributed Cryptography With General Trust. Proc. Stabilization, Safety, and Security of Distributed Systems (SSS), 536-551. https://doi.org/10.1007/978-3-031-44274-2_40

Literature

UNIVERSITÄT BERN

- Model 3: Asymmetric trust
  - [AZ21] Cachin, C., & Zanolini, L. (2021). Asymmetric Asynchronous Byzantine Consensus. Proc. ESORICS Workshops on Data Privacy Management (DPM), Cryptocurrencies and Blockchain Technology (CBT) LNCS 13140, 192–207. https://doi.org/10.1007/978-3-030-93944-1_13
  - [ACTZ24] Alpos, O., Cachin, C., Tackmann, B., & Zanolini, L. (2024). Asymmetric Distributed Trust. Distributed Computing, 37, Online. https://doi.org/10.1007/s00446-024-00469-1
  - [CLZ22] Cachin, C., Losa, G., & Zanolini, L. (2022). Quorum Systems in Permissionless Proc. 26th International Conference on Principles of Distributed Systems (OPODIS), 17:1–17:22. https://doi.org/10.4230/LIPIcs.OPODIS.2022.17

Literature

- Model 3: Asymmetric trust
  - [ACM21] Amores-Sesar, I., Cachin, C., & Mićić, J. (2021). Security Analysis of Ripple Consensus. Proc. 24th International Conference on Principles of Distributed Systems (OPODIS), 10:1–10:16. https://doi.org/10.4230/LIPIcs.OPODIS.2020.10

Literature

- Model 4: Unstructured, probabilistic voting
  - [ACT22] Amores-Sesar, I., Cachin, C., & Tedeschi, E. (2022). When is Spring coming? A Security Analysis of Avalanche Consensus. Proc. 26th International Conference on Principles of Distributed Systems (OPODIS), 10:1–10:22. https://doi.org/10.4230/LIPIcs.OPODIS.2022.10

  - [ACS24] Amores-Sesar, I., Cachin, C., & Schneider, P. (2024). An Analysis of Avalanche Consensus. In Y. Emek (Ed.), Proc. Structural Information and Communication Complexity (SIROCCO) (Vol. 14662, pp. 27–44). Springer. https://doi.org/10.1007/978-3-031-60603-8_2

Literature

- Models 5-7: Stake based
  - [B21] Bürk, T. (2022). Blockchain consensus protocols based on stake. Master thesis, Institute of Computer Science, University of Bern. https://crypto.unibe.ch/archive/theses/2021.msc.timo.buerk.pdf
  - [AC23] Alpos, O., Cachin, C., Holmgaard Kamp, S., & Buus Nielsen, J. (2023). Practical Large-Scale Proof-Of-Stake Asynchronous Total-Order Broadcast. Proc. 5th Conference on Advances in Financial Technologies (AFT), 31:1–31:22. https://doi.org/10.4230/LIPIcs.AFT.2023.31

Literature

- Model 9
  - [ACP21] Amores-Sesar, I., Cachin, C., & Parker, A. (2021). Generalizing Weighted Trees: A Bridge from Bitcoin to GHOST. Proc. 3rd ACM Conference on Advances in Financial Technologies (AFT), 156–169. https://doi.org/10.1145/3479722.3480995

  - [ACLVZ22] Azouvi, S., Cachin, C., Le, D. V., Vukolic, M., & Zanolini, L. (2022). Modeling Resources in Permissionless Longest-Chain Total-Order Broadcast. Proc. 26th International Conference on Principles of Distributed Systems (OPODIS), 19:1–19:23. https://doi.org/10.4230/LIPIcs.OPODIS.2022.19