Fast and Secure Global Payments with Stellar

Marta Lokhava, Giuliano Losa, David Mazières, Graydon Hoare,
Nicolas Barry, Eli Gafni, Jonathan Jove, Rafał Malinowsky, and Jed McCaleb
Stellar Development Foundation

Presented By: Sailendra Akash Bonagiri, Naresh Kumar Kaushal, Vikraman Senthil Kumar, Anuraag Velamati

UCDAVIS
COLLEGE OF ENGINEERING

What is Stellar?

Why do we need Stellar?

UCDAVIS COLLEGE OF ENGINEERING

What is Stellar?

Stellar is a blockchain-based payment network specifically designed to facilitate innovation and competition in international payments.

Why do we need Stellar?

Motivation

- International Payments are slow and expensive due to multihop payment routing through heterogeneous banking systems

Example: Alice wants to send $0.5 from the U.S to Bob who is in the neighbouring country Mexico. The user will have to pay $9 for such a transaction given all the fees with multi hops in between and a bilateral agreement brokered by the countries' central banks could only reduce the underlying bank cost to $0.67

- There is also Latency issues involved with international payments

UC DAVIS
COLLEGE OF ENGINEERING

Motivation

- An international payment generally takes days. If a payment is to be made urgently in case of an emergency, it stands a big trouble.

- In countries with no proper banking systems or fees are very high, citizens rely on modes of transport such as bus, ship etc. to make payments.

- This has huge risk associated along with latency or inconvenience

Motivation

- The key innovation here is a secure transaction mechanism across untrusted intermediaries, using a new Byzantine agreement protocol called SCP - Stellar Consensus Protocol.
- With SCP, each institution specifies other institutions with which to remain in agreement.
- Through the global interconnectedness of the financial system, the whole network then agrees on atomic transactions spanning arbitrary institutions, with no solvency or exchange-rate risk from intermediary asset issuers or market makers

UCDAVIS COLLEGE OF ENGINEERING

What is Stellar?

Stellar is a blockchain-based payment network specifically designed to facilitate innovation and competition in international payments.

Why do we need Stellar?

What is Stellar?

Stellar is a blockchain-based payment network specifically designed to facilitate innovation and competition in international payments.

Why do we need Stellar?

- Stellar can directly transfer digital money anywhere in the world in seconds
- Stellar is the first system to meet all three of the following goals - Open Membership, Issuer-enforced finality, Cross-issuer atomicity

UCDAVIS COLLEGE OF ENGINEERING

What are these Goals?

- **Open membership** – Anyone can issue currency-backed digital tokens that can be exchanged among users.
- **Issuer-enforced finality** – A token's issuer can prevent transactions in the token from being reversed or undone.
- **Cross-issuer atomicity** – Users can atomically exchange and trade tokens from multiple issuers

Issues with achieving these Goals

Famous Payment Gateways:

- Any company can unilaterally offer a product such as Paypal, Venmo etc, and ensure the finality of payments in the virtual currencies they have created. Unfortunately, transacting atomically across these currencies is impossible. End users cannot send Venmo dollars to Paypal users and vice versa.

UCDAVIS
COLLEGE OF ENGINEERING

Issues with achieving these Goals

Closed systems:

- In particular, a number of countries have efficient domestic payment networks, typically overseen by a universally trusted regulatory authority. However, membership is limited to a closed set of chartered banks and the networks are limited to the reach of a country’s regulatory authority. Here Goals Issuer enforced finality and Cross issuer atomicity can be ensured.

UCDAVIS COLLEGE OF ENGINEERING

Issues with achieving these Goals

Mined Blockchains:

- The key idea of these blockchains is to create a new cryptocurrency with which to reward people for making settled transactions hard to revert. Unfortunately, this means token issuers do not control transaction finality. If software errors cause transactions history to be reorganized, or when the spoils of defrauding people exceed the cost of reorganizing history, issuers may be liable for tokens they have already redeemed for real-world money. Here the goals Open membership and Cross-Issuer atomicity can be achieved but not Issuer enforced finality.

How does Stellar solve this?

- Stellar natively facilitates markets with tokens from different issuers.

- Specifically, anyone can issue a token, the blockchain provides a built-in orderbook for trade between any pair of tokens, and users can issue path payments that atomically trade across several currency pairs while guaranteeing an end-to-end limit price

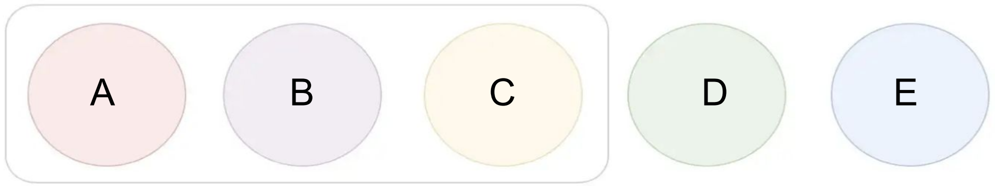
How does Stellar solve this?

- To enforce transaction finality, Stellar introduces a new Byzantine agreement protocol, SCP (Stellar Consensus Protocol), through which token issuers designate specific validator servers.

- So long as no one compromises an issuer’s validators (and the underlying digital signatures and cryptographic hashes remain secure), the issuer knows exactly which transactions have occurred and avoids the risk of losses from blockchain history reorganization

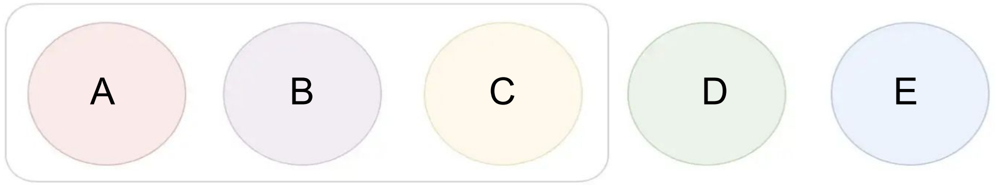
Terminologies

1. Quorum Slice : A set of nodes which can convince another specific node to agree.



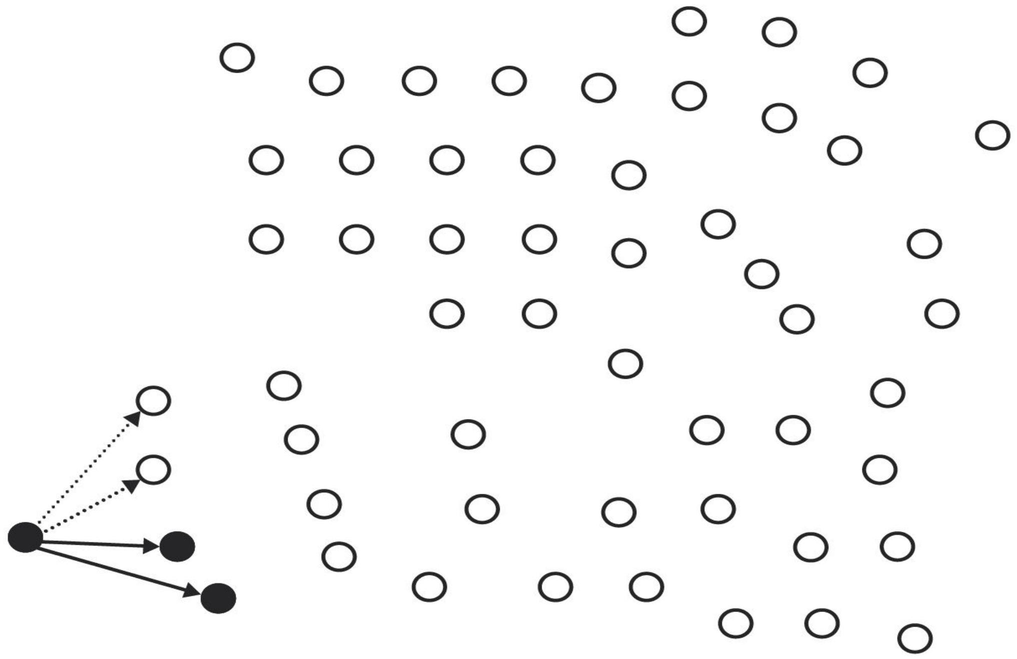
Terminologies

1. Quorum Slice : A set of nodes which can convince another specific node to agree.



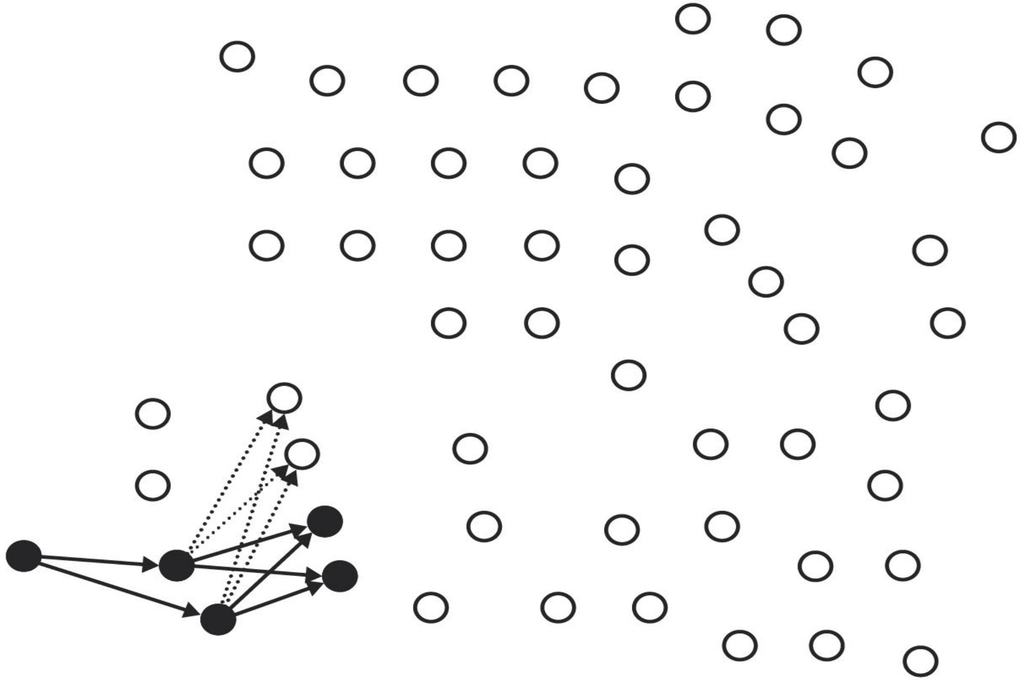
2. Quorum : A quorum is a set of nodes sufficient to reach agreement in the whole network. We can build a quorum by transitive closure of starting node's quorum slice.

How to form quorum from quorum slice


Choose just one quorum slice at each step.

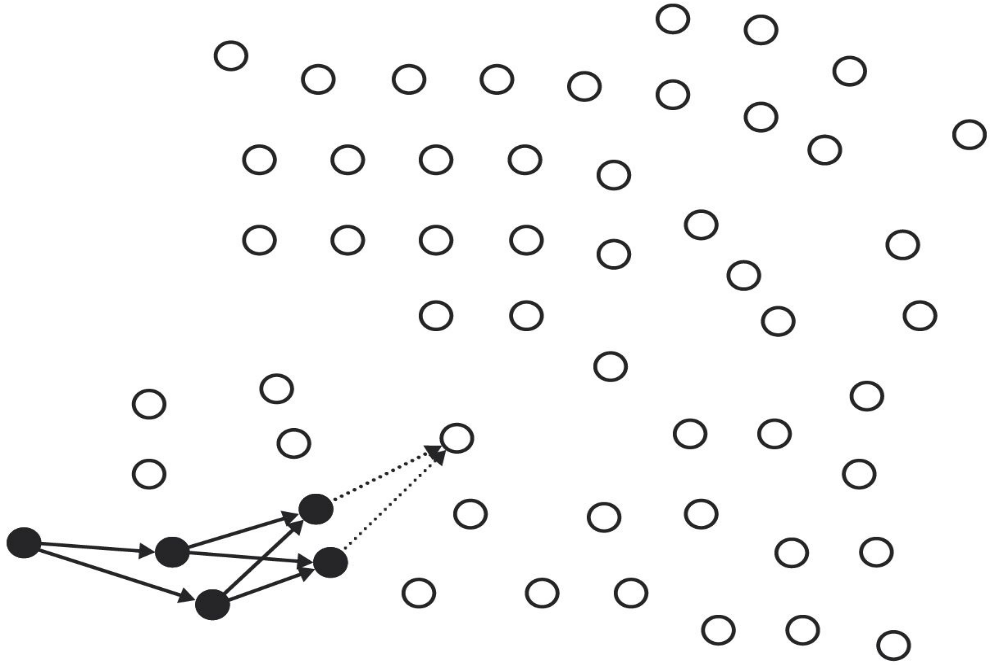
Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

How to form quorum from quorum slice



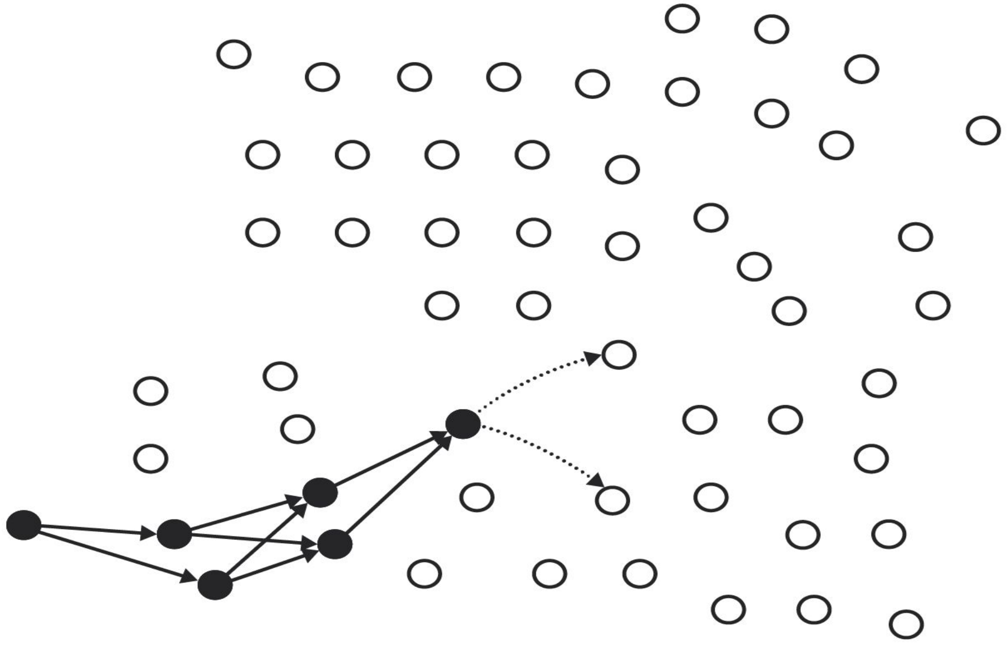
Ref : Understanding the stellar consensus protocol by Bob Glickstein [Link]

How to form quorum from quorum slice


...then add the members of those nodes' slices.

Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

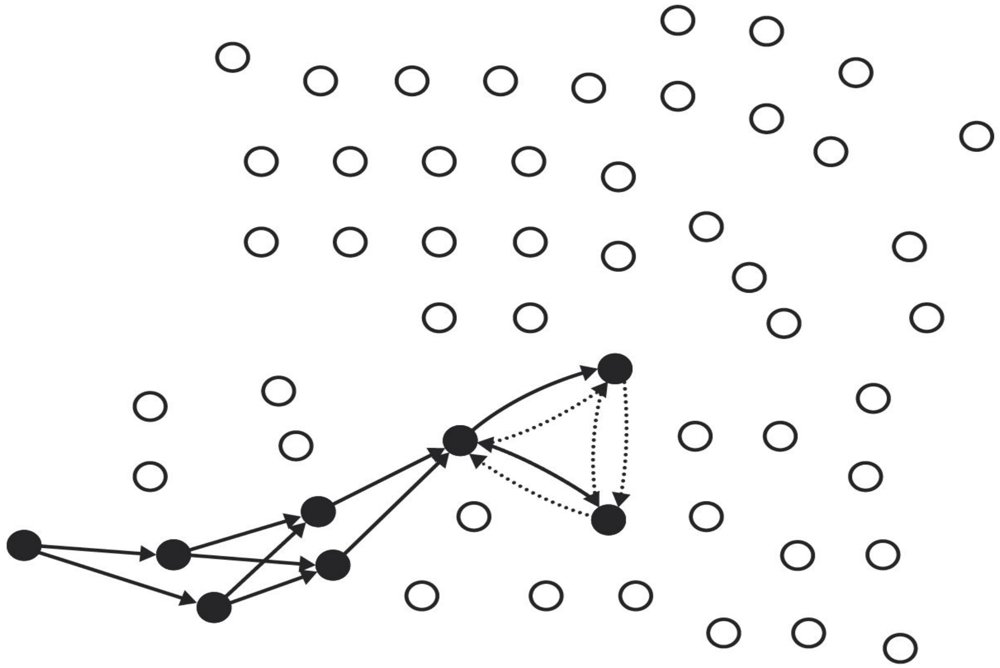
How to form quorum from quorum slice



Continue until there are no nodes to add.

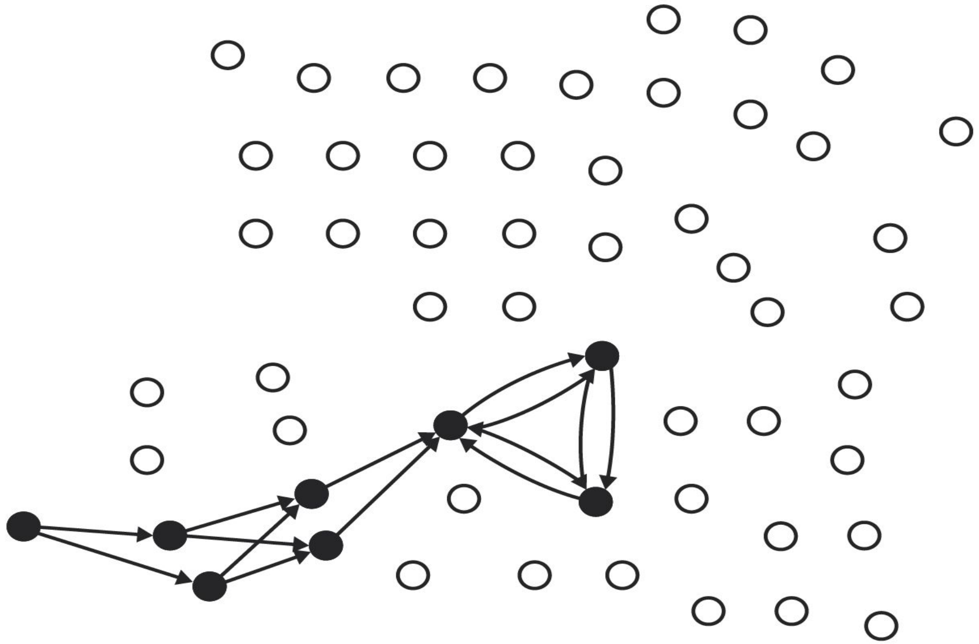
Ref : Understanding the stellar consensus protocol by Bob Glickstein [Link]

How to form quorum from quorum slice



Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

How to form quorum from quorum slice



No remaining nodes to add. This is a quorum.

Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

How does a node know the membership of another node’s quorum slices?

UC DAVIS
COLLEGE OF ENGINEERING

How does a node know the membership of another node’s quorum slices?

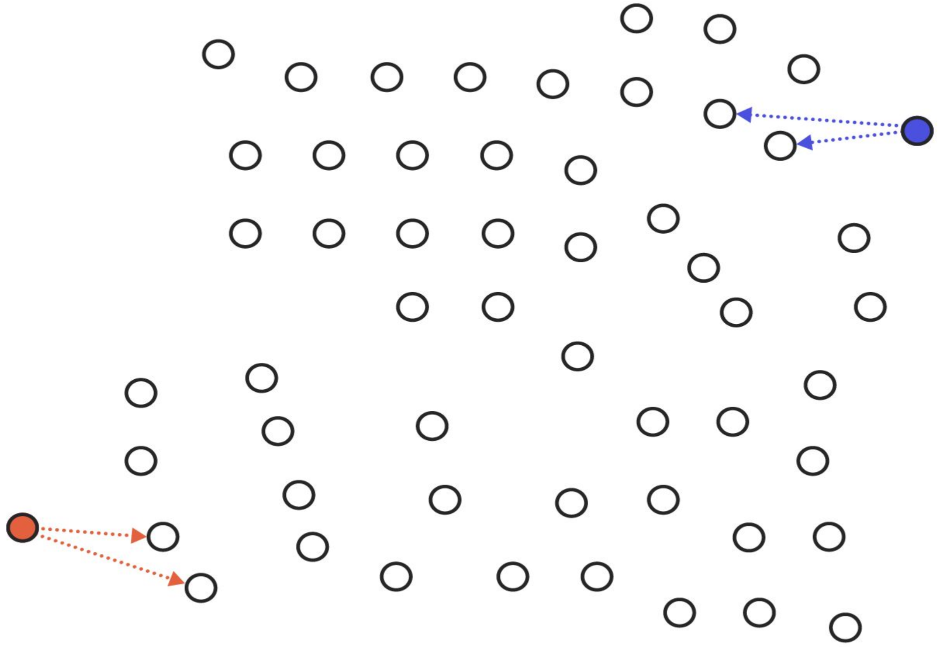
By Broadcasting a message : “I am node A” my quorum slices are Q(A) and “I am voting for X”

Quorum Intersection

When 2 quorums intersect they result in having overlapping nodes known as quorum intersection.

Let’s see how does a quorum intersection looks like……..

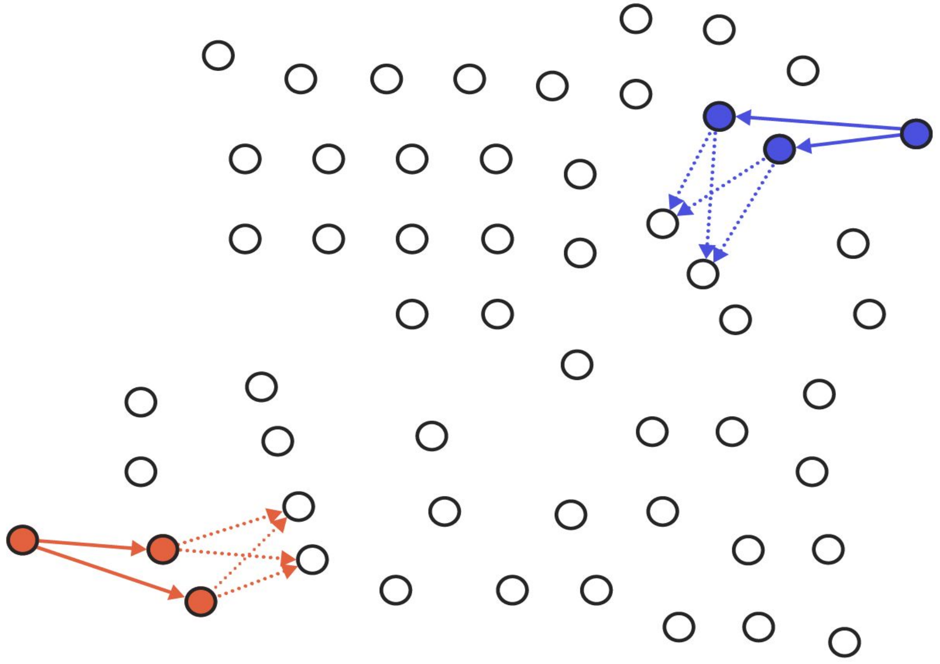
Quorum Intersection



If the network has “quorum intersection”...

Ref : Understanding the stellar consensus protocol by Bob Glickstein [Link]

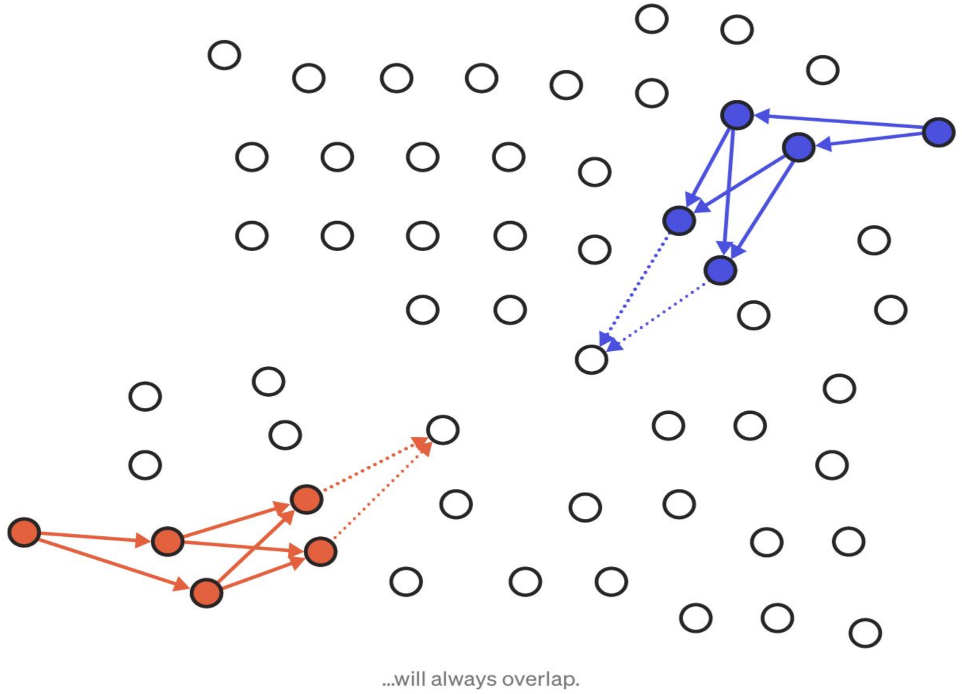
Quorum Intersection



...then any two quorums you might construct...

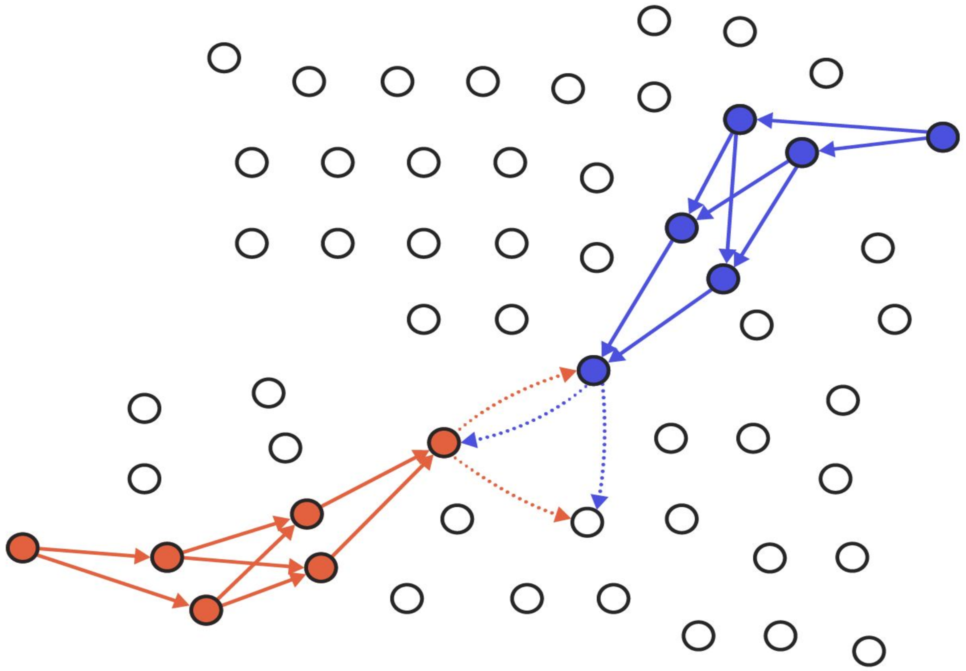
Ref : Understanding the stellar consensus protocol by Bob Glickstein [Link]

Quorum Intersection



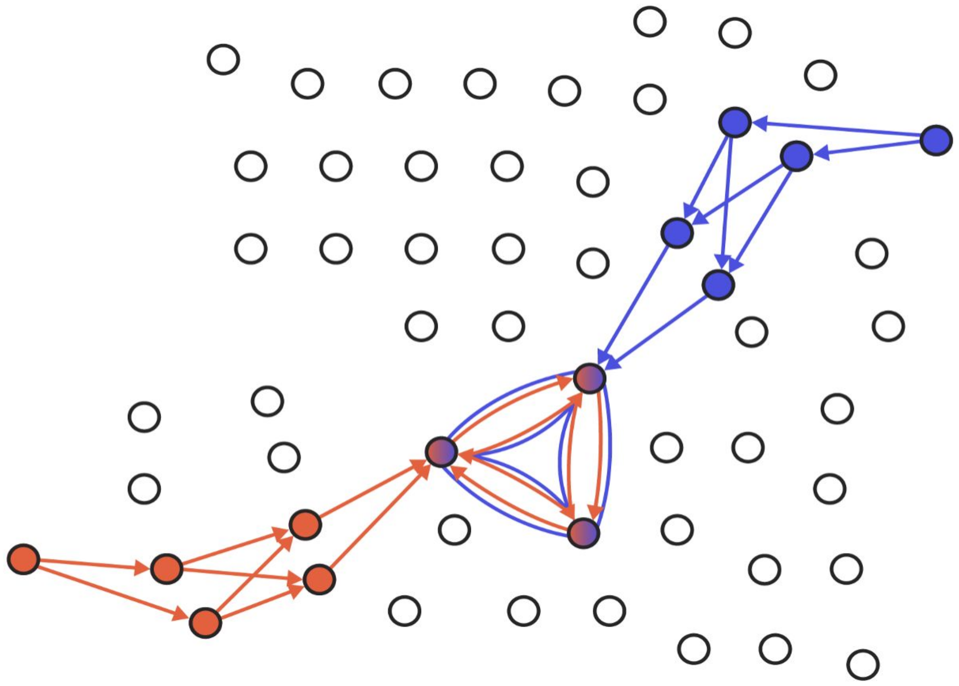
Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

Quorum Intersection



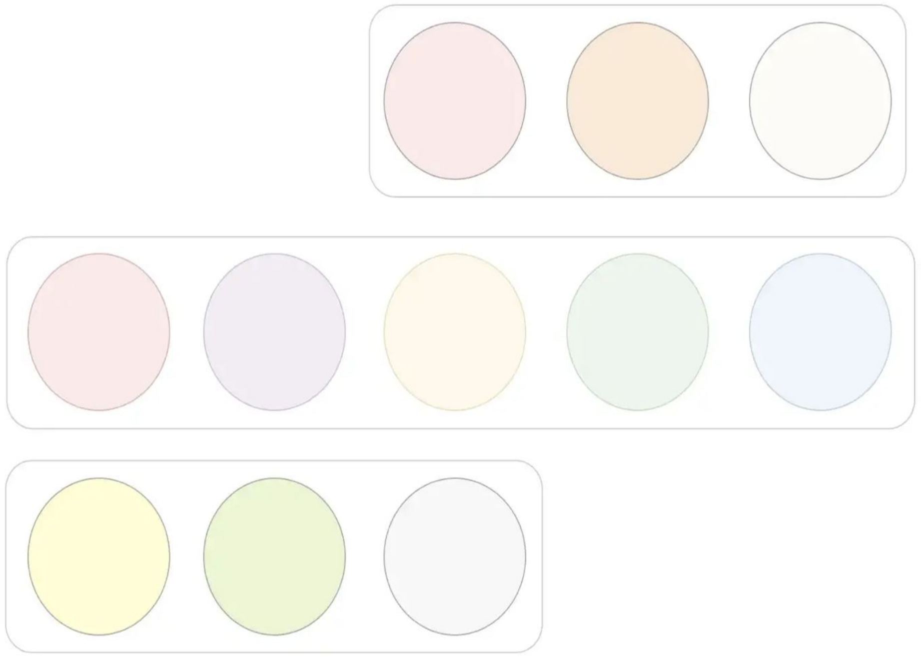
Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

Quorum Intersection



Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

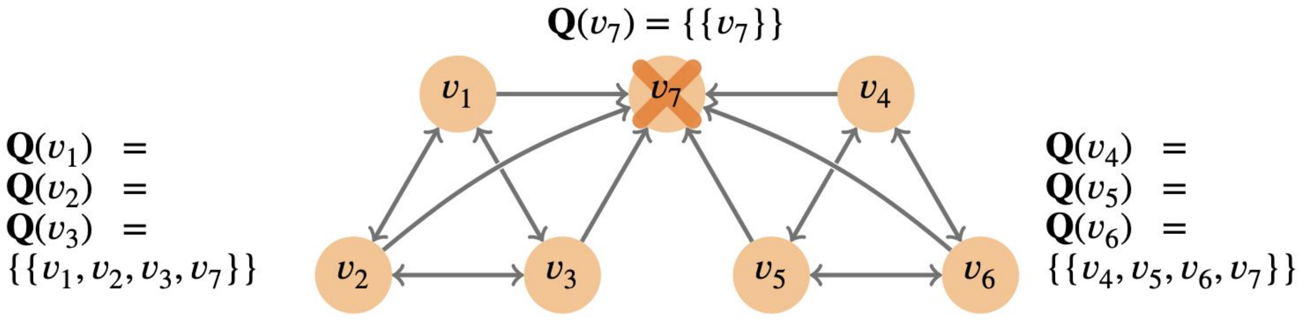
Disjoint Quorums


3 Disjoint Quorums.

Disjoint quorums are undesirable because each of them can independently and simultaneously agree on contradictory transactions, thereby undermining overall consensus and leads to *safety* concerns.

Intertwined nodes

Non-faulty nodes v1 and v2 are intertwined when every quorum of v1 intersects every quorum of v2 in at least one non-faulty node.



$\mathbf{Q}(v_7) = \{\{v_7\}\}$
$\mathbf{Q}(v_1) = \mathbf{Q}(v_2) = \mathbf{Q}(v_3) = \{\{v_1, v_2, v_3, v_7\}\}$
$\mathbf{Q}(v_4) = \mathbf{Q}(v_5) = \mathbf{Q}(v_6) = \{\{v_4, v_5, v_6, v_7\}\}$
v_1
v_7
v_4
v_2
v_3
v_5
v_6

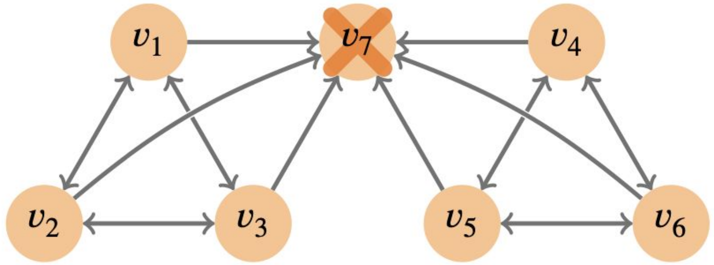
Fig. 7. Ill-behaved node v_7 can undermine quorum intersection.

Intertwined nodes

Non-faulty nodes v1 and v2 are intertwined when every quorum of v1 intersects every quorum of v2 in at least one non-faulty node.

$$\mathbf{Q}(v_7) = \{\{v_7\}\}$$

$$\mathbf{Q}(v_1) = \mathbf{Q}(v_2) = \mathbf{Q}(v_3) = \{\{v_1, v_2, v_3, v_7\}\}$$



$$\mathbf{Q}(v_4) = \mathbf{Q}(v_5) = \mathbf{Q}(v_6) = \{\{v_4, v_5, v_6, v_7\}\}$$

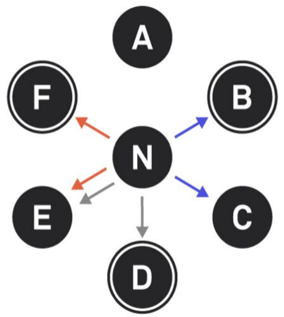
Fig. 7. Ill-behaved node v_7 can undermine quorum intersection.

An FBA protocol can ensure agreement only between intertwined nodes.

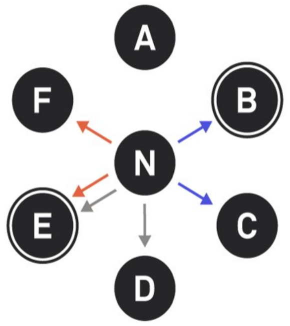
Blocking set

A set (B) of nodes containing at least one member of each of N’s slices.

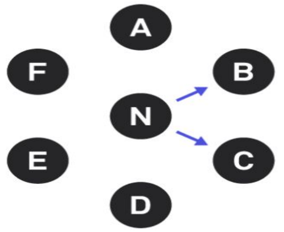
A blocking set with all the bad nodes can cost the *liveliness* of the network.



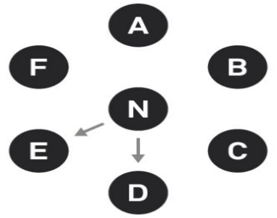
B-D-F is a blocking set for N: it includes one node from each of N’s s



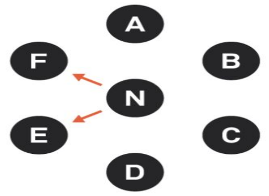
B-E is also a blocking set for N, because E appears in two of N’s slices.



N’s Quorum Slice 1



N’s Quorum Slice 2



N’s Quorum Slice 3

Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

UCDAVIS COLLEGE OF ENGINEERING

Federated Voting

- Node’s opinion on a statement depends on the opinion of it’s quorum slice

Federated Voting

- **Node's opinion on a statement depends on the opinion of it's quorum set**
- **A node might say 4 things about statement A**
  - I don't know anything about A and have no opinion

Federated Voting

- **Node’s opinion on a statement depends on the opinion of it’s quorum set**
- **A node might say 4 things about statement A**
  - I don’t know anything about A and have no opinion
  - I voted for A, its valid but I don’t know if it’s safe to act on it yet

Federated Voting

- **Node's opinion on a statement depends on the opinion of it's quorum set**
- **A node might say 4 things about statement A**
  - I don't know anything about A and have no opinion
  - I voted for A, its valid but I don't know if it's safe to act on it yet
  - I accept **A**, because node in my quorum slice support this statement, but I don't know if it's safe to act on it yet.

UCDAVIS
COLLEGE OF ENGINEERING

Federated Voting

- **Node's opinion on a statement depends on the opinion of it's quorum set**
- **A node might say 4 things about statement A**
    - I don't know anything about A and have no opinion
    - I voted for A, its valid but I don't know if it's safe to act on it yet
    - I accept **A**, because node in my quorum slice support this statement, but I don't know if it's safe to act on it yet.
    - I confirm **A**, **it is safe to act on it.** Even if every node in a quorum slice has not yet confirmed **A,** they will not be able to confirm anything else but **A.**

UCDAVIS COLLEGE OF ENGINEERING

Steps in Federated Byzantine Agreement

- **Voting** : Nodes optionally begin by casting a vote for some value V.

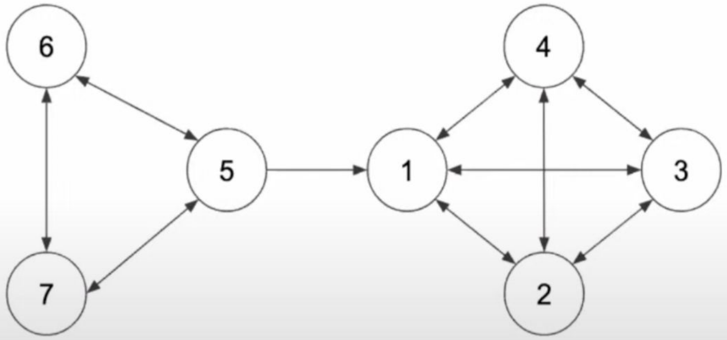
- **Accepting** : There are 2 ways for a node to accept a statement ‘a’
  (a) There exist a quorum slice that voted or accepted ‘a’
  (b) Each member in v blocking set claims to accept ‘a’

Statement (b) allows the node to vote for statement ‘a’ and then later accept a contradictory one.

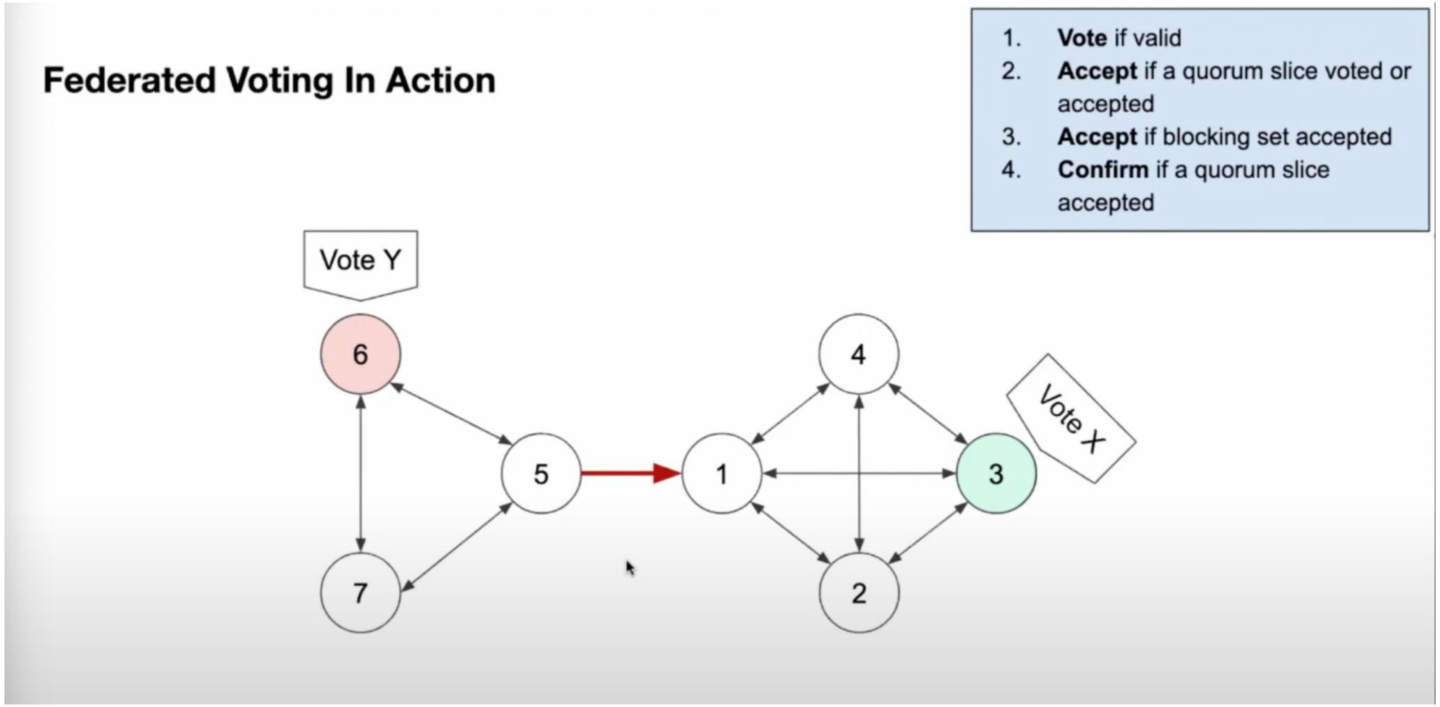
- **Confirming** : When a node sees the entire quorum has accepted a statement ‘a’ then it confirms that statement.

Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted



Ref: Stellar development foundation *Link*



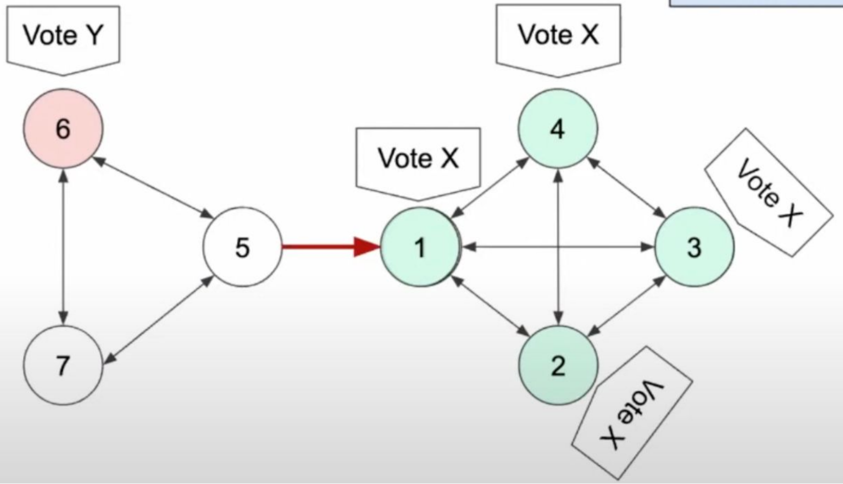
Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

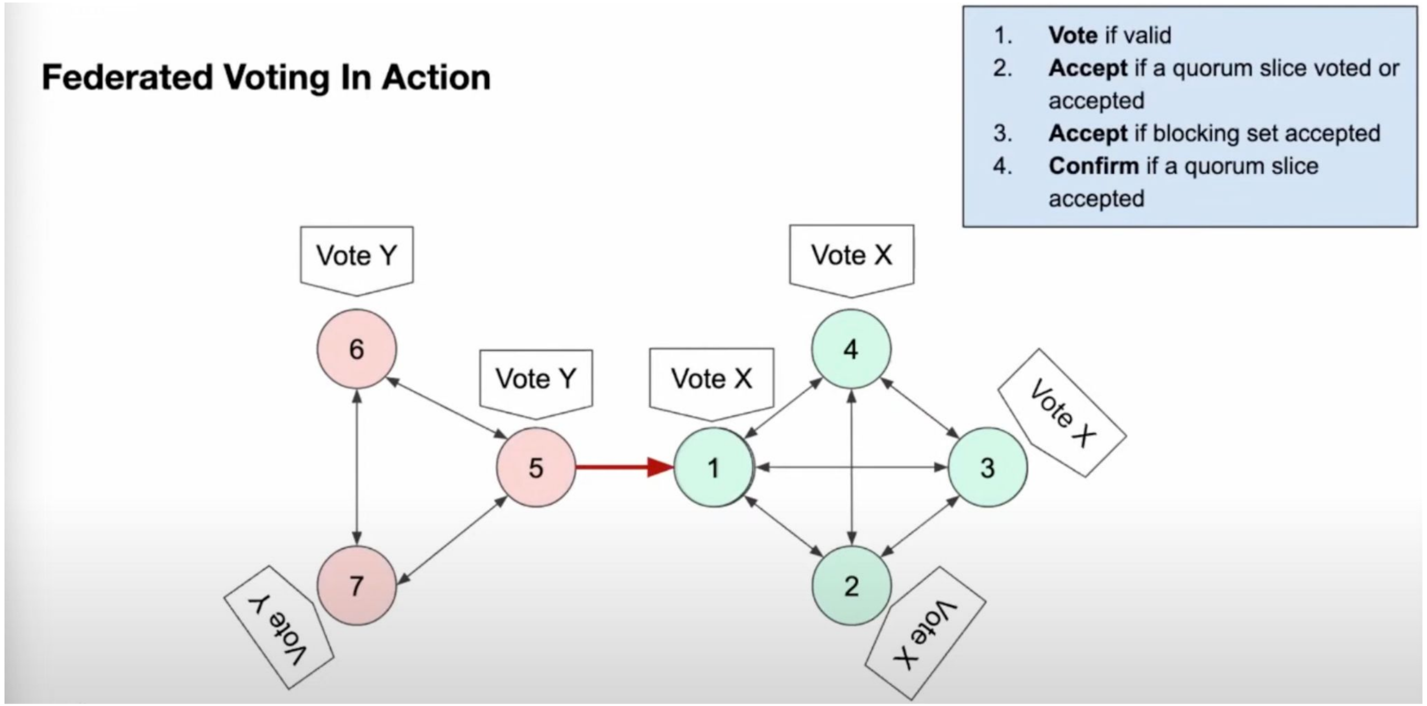
Ref: Stellar development foundation Link

Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted



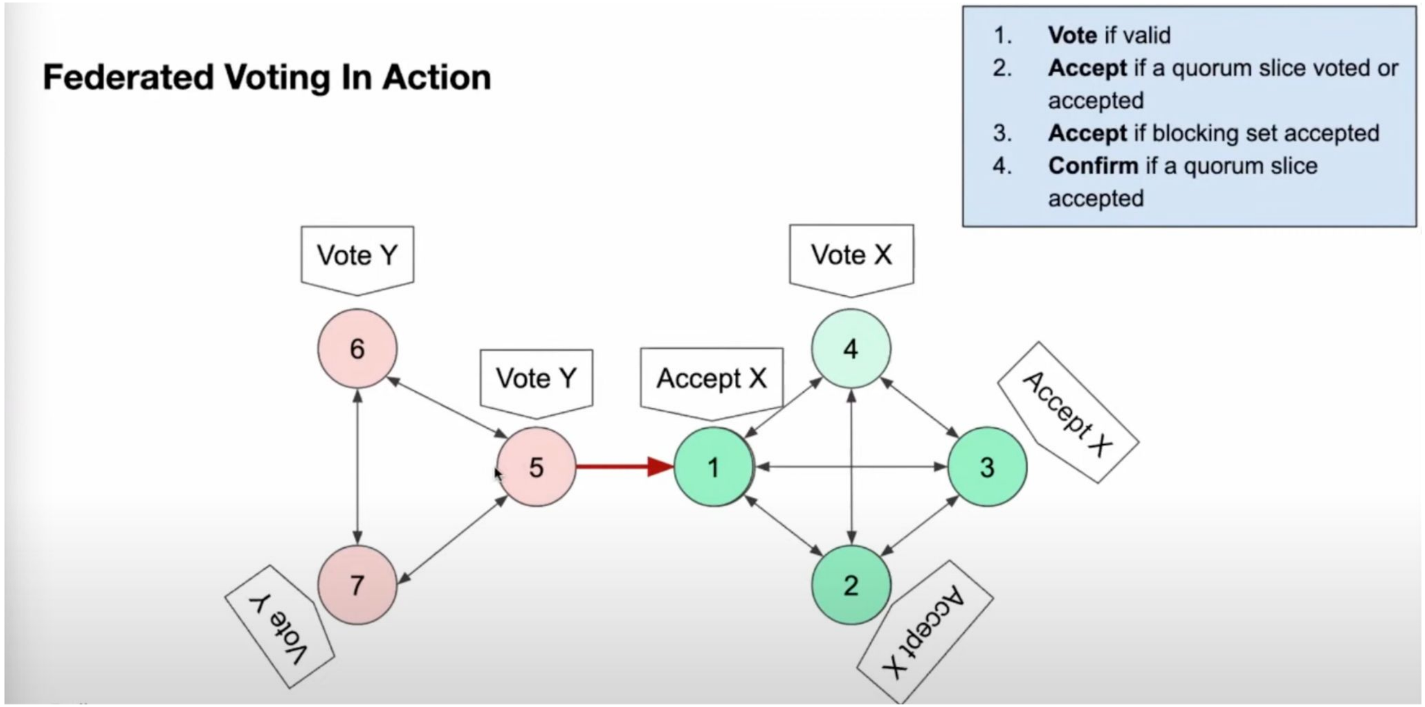
Ref: Stellar development foundation Link



Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

Ref: Stellar development foundation Link



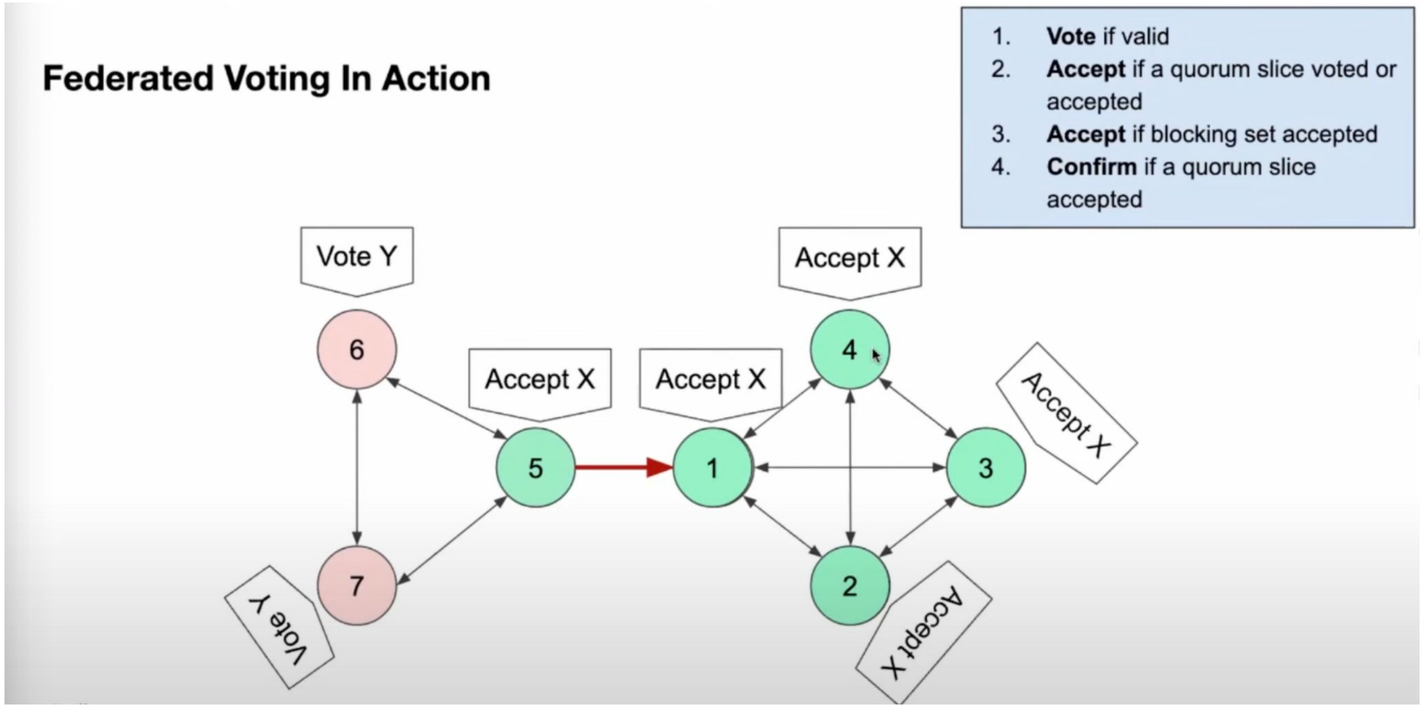
Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

Ref: Stellar development foundation Link

Why didn’t Node 4 accept X?

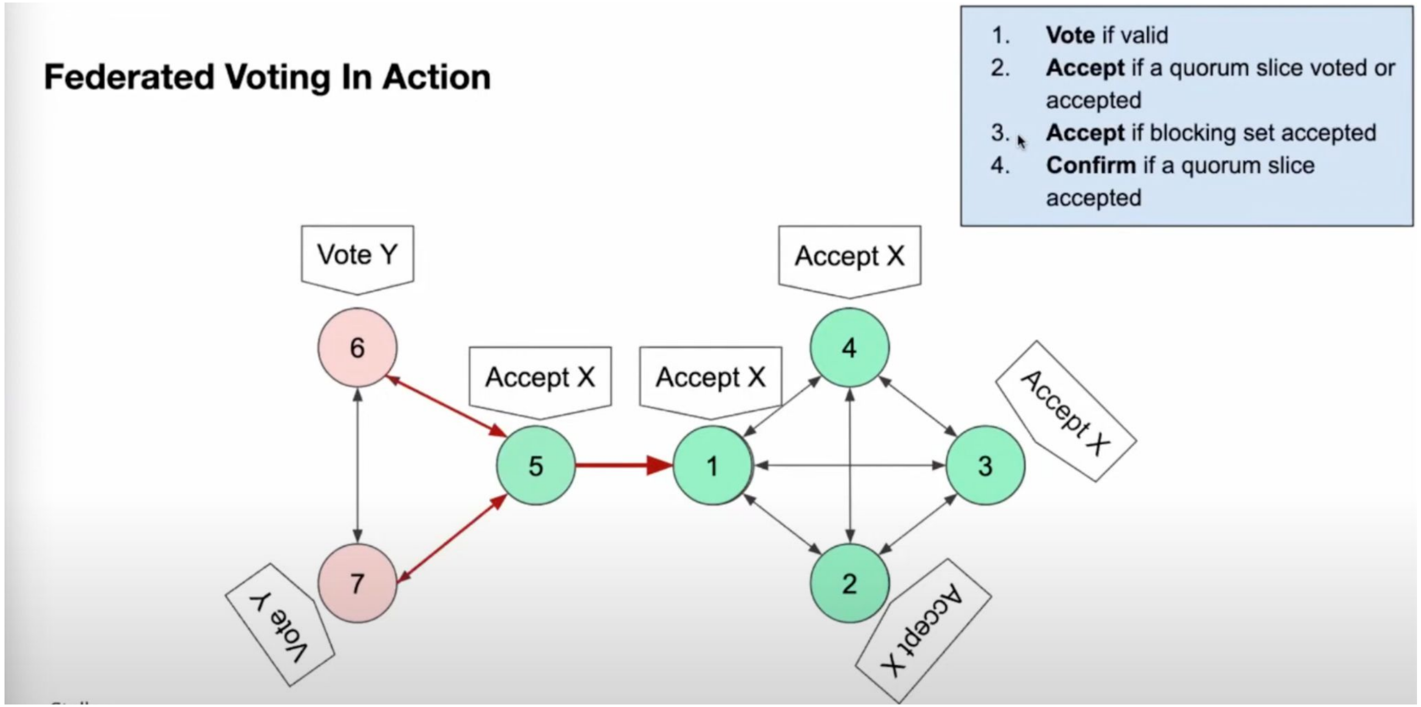
As Federal Byzantine Agreement is an asynchronous protocol, so messages can be delayed to reach their destination but reaching the destination is certain.



Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

Ref: Stellar development foundation Link



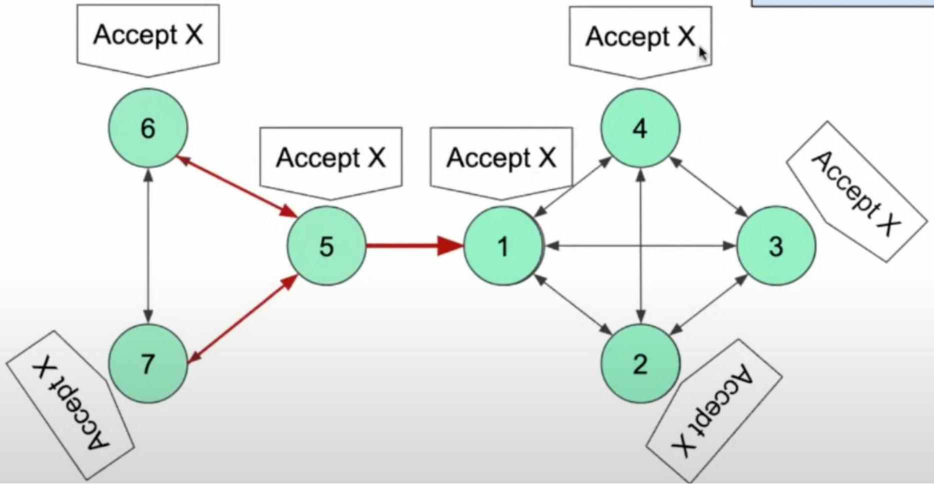
Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

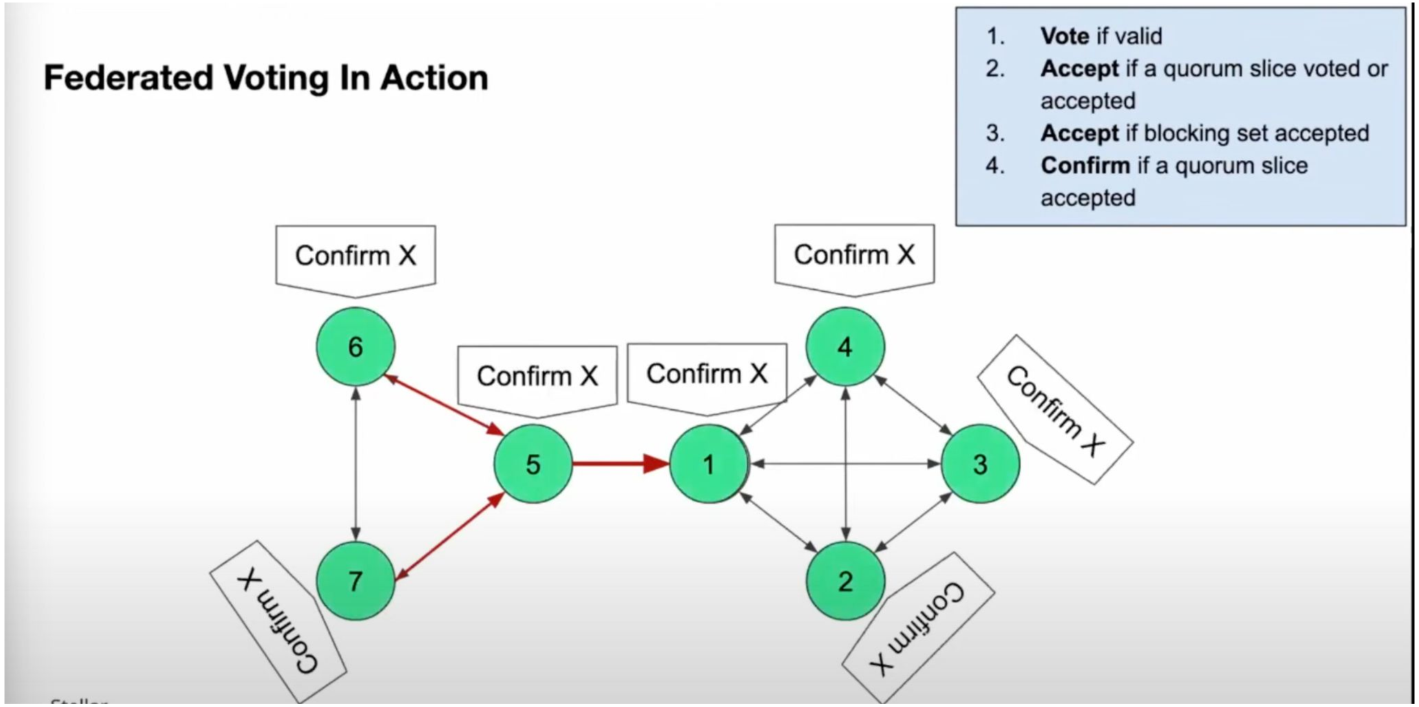
Ref: Stellar development foundation [Link]

Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted



Ref: Stellar development foundation [Link]()

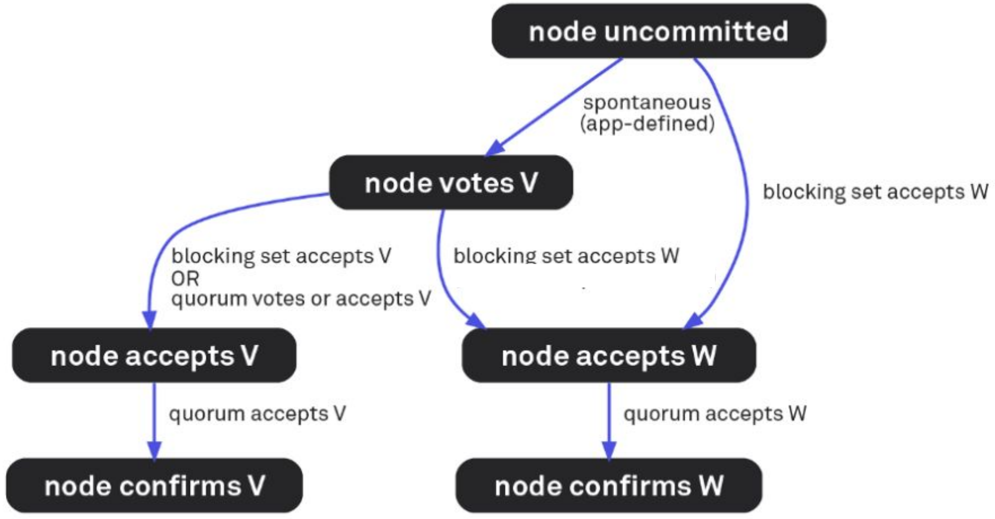


Federated Voting In Action

1. **Vote** if valid
2. **Accept** if a quorum slice voted or accepted
3. **Accept** if blocking set accepted
4. **Confirm** if a quorum slice accepted

Ref: Stellar development foundation *Link*

Recap of what we have learnt so far



Federated voting.

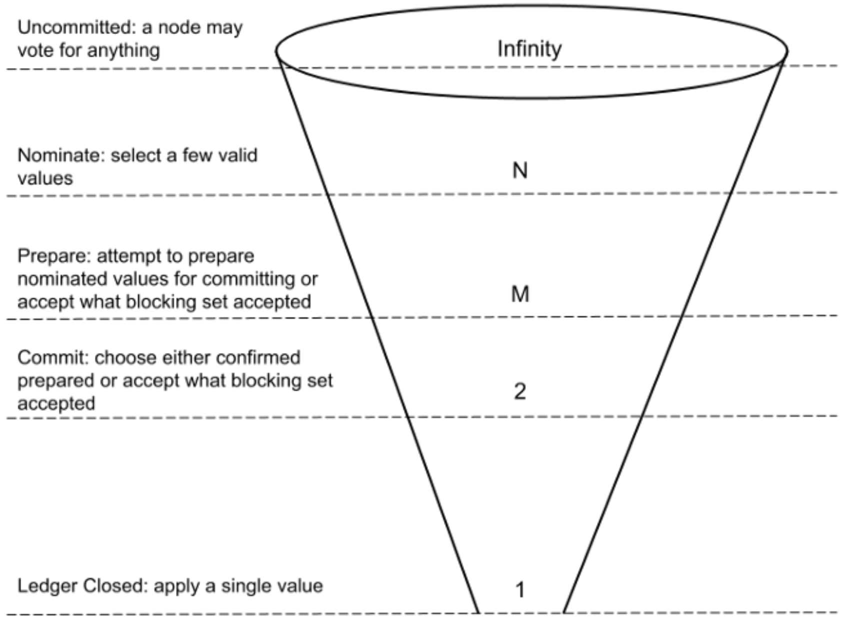
Ref : Understanding the stellar consensus protocol by Bob Glickstein Link

What is Stellar consensus protocol?

SCP is a Federation-based Byzantine Agreement (FBA) protocol which allows open membership instead of requiring closed membership

Federal Voting in SCP

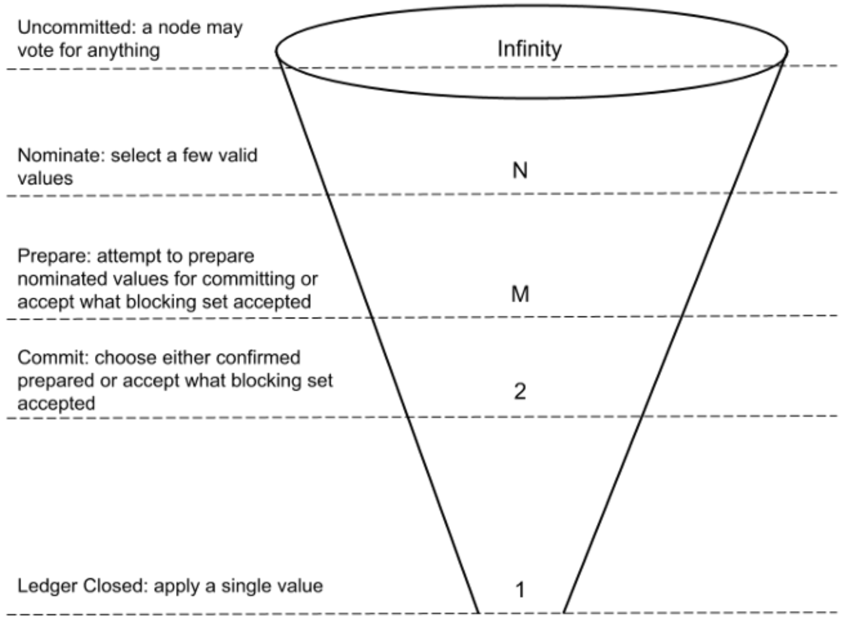
Uncommitted: A node starts with a *possibility of agreeing on any value* (in the context of Stellar, a value is a *transaction set*)


Uncommitted: a node may vote for anything
Infinity
Nominate: select a few valid values
N
Prepare: attempt to prepare nominated values for committing or accept what blocking set accepted
M
Commit: choose either confirmed prepared or accept what blocking set accepted
2
Ledger Closed: apply a single value
1

Ref : Intuitive Stellar Consensus Protocol Link

Federal Voting in SCP

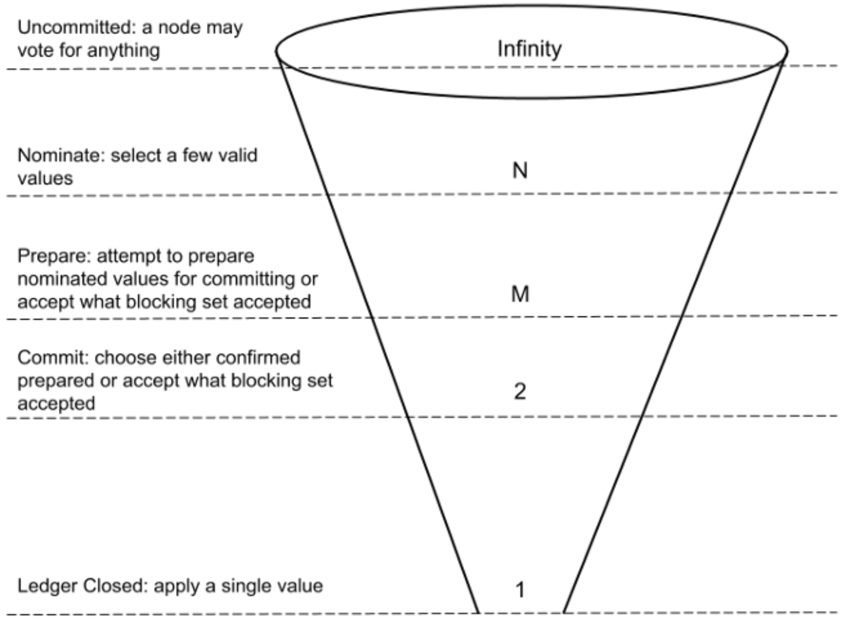
Nominate: statements aim to select an (ideally) small number N of candidate valid values.


Uncommitted: a node may vote for anything
Infinity
Nominate: select a few valid values
N
Prepare: attempt to prepare nominated values for committing or accept what blocking set accepted
M
Commit: choose either confirmed prepared or accept what blocking set accepted
2
Ledger Closed: apply a single value
1

Ref : Intuitive Stellar Consensus Protocol Link

Federal Voting in SCP

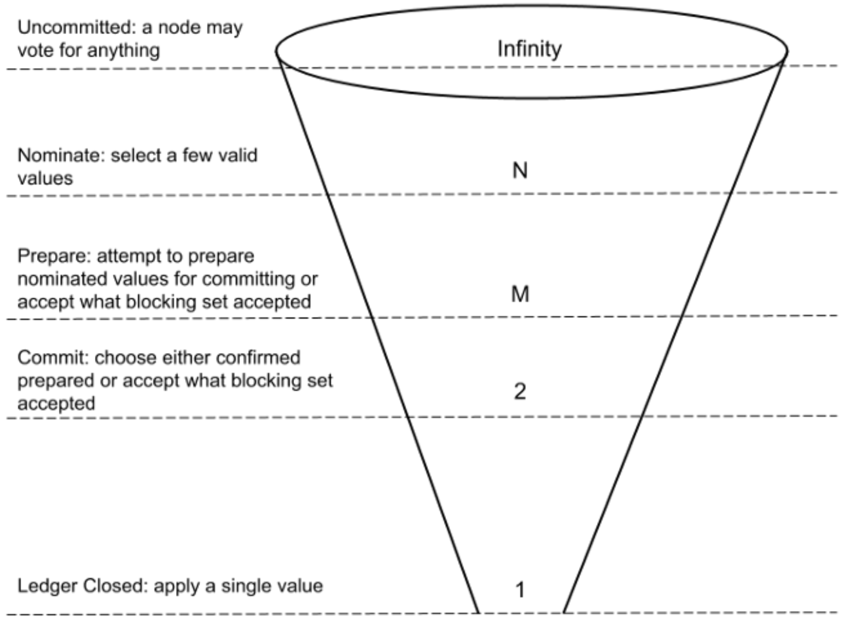
Prepare: statements attempt to ensure a subset M (much smaller than N) of that finite set (i.e. any values that got confirmed nominated) can be committed


Uncommitted: a node may vote for anything
Infinity
Nominate: select a few valid values
N
Prepare: attempt to prepare nominated values for committing or accept what blocking set accepted
M
Commit: choose either confirmed prepared or accept what blocking set accepted
2
Ledger Closed: apply a single value
1

Ref : Intuitive Stellar Consensus Protocol
Link

Federal Voting in SCP

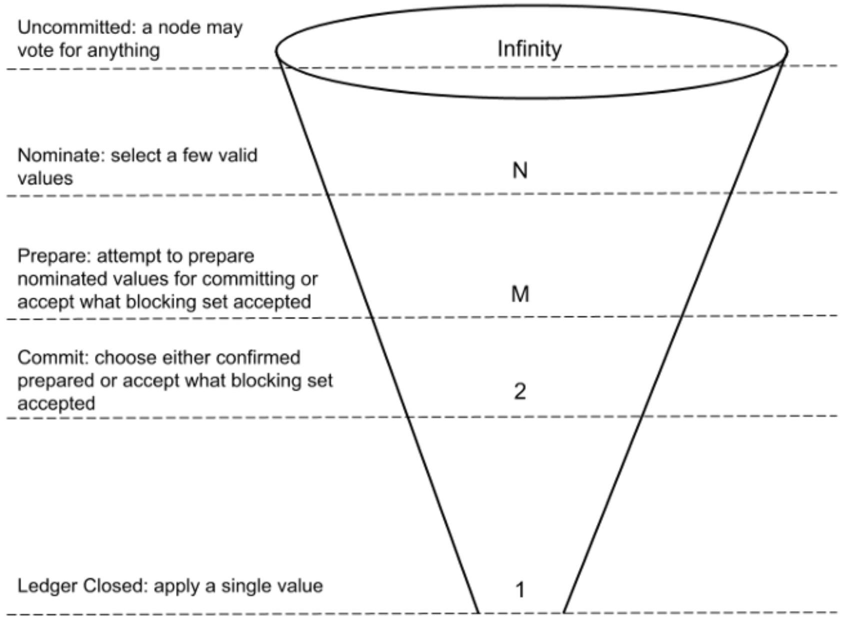
Commit: Statements either proceed to commit a confirmed prepared value, or commit what a node's blocking set committed


Uncommitted: a node may vote for anything
Infinity
Nominate: select a few valid values
N
Prepare: attempt to prepare nominated values for committing or accept what blocking set accepted
M
Commit: choose either confirmed prepared or accept what blocking set accepted
2
Ledger Closed: apply a single value
1

Ref : Intuitive Stellar Consensus Protocol Link

Federal Voting in SCP

Ledger closed: This is where we know the protocol has agreed on a single value



Uncommitted: a node may vote for anything
Infinity
Nominate: select a few valid values
N
Prepare: attempt to prepare nominated values for committing or accept what blocking set accepted
M
Commit: choose either confirmed prepared or accept what blocking set accepted
2
Ledger Closed: apply a single value
1

Ref : Intuitive Stellar Consensus Protocol Link

CAP THEOREM

CAP Theorem states that out of the three topics: consistency, availability and partition tolerance, only two can be guaranteed

CAP THEOREM

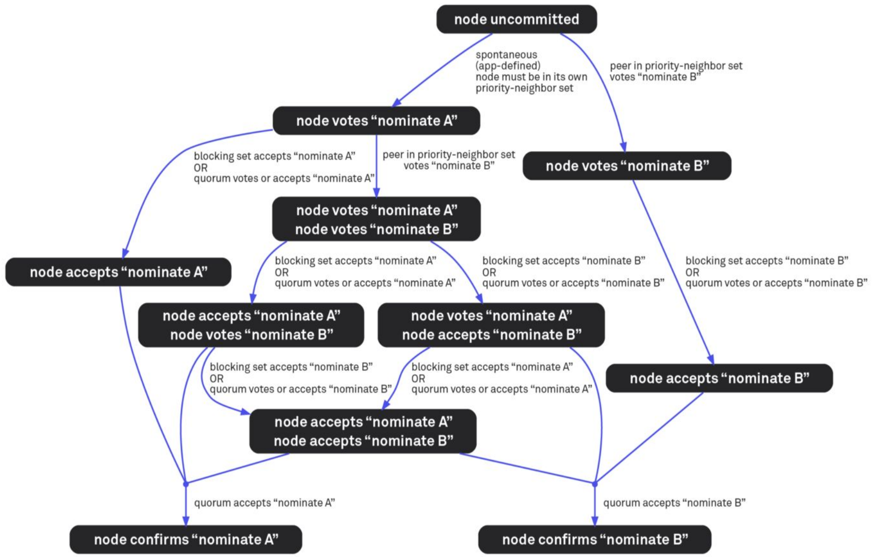
- SCP is designed to ensure consistency
- The cost of this consistency reduces availability

Nomination Protocol

- Federated voting on statements like "Nominate transaction set x"
- A node may vote to nominate any transaction set, as long as it's valid
- Any confirmed nomination is added to the candidate set
- A node stops voting for new values, as soon as it confirms its first nomination. It may still accept and confirm whatever its blocking set accepted/confirmed
- A node combines candidates into the composite value, and starts the ballot protocol on it

Nomination Protocol

- Federated voting on statements like "Nominate transaction set x".
- Any confirmed nomination is added to the candidate set.
- A node stops voting for new values, as soon as it confirms its first nomination. It may still accept and confirm whatever its blocking set accepted/confirmed.
- A node combines candidates into the composite value, and starts the ballot protocol on it.



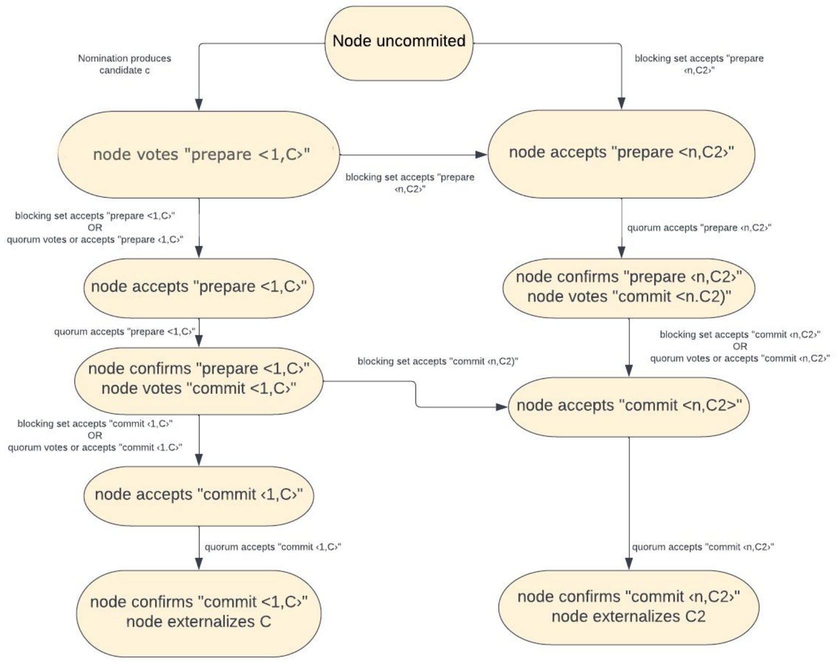
node uncommitted
spontaneous (app-defined) node must be in its own priority-neighbor set
peer in priority-neighbor set votes "nominate B"
node votes "nominate A"
blocking set accepts "nominate A" OR quorum votes or accepts "nominate A"
peer in priority-neighbor set votes "nominate B"
node votes "nominate B"
node votes "nominate A" node votes "nominate B"
node accepts "nominate A"
blocking set accepts "nominate A" OR quorum votes or accepts "nominate A"
blocking set accepts "nominate B" OR quorum votes or accepts "nominate B"
blocking set accepts "nominate B" OR quorum votes or accepts "nominate B"
node accepts "nominate A" node votes "nominate B"
node votes "nominate A" node accepts "nominate B"
blocking set accepts "nominate B" OR quorum votes or accepts "nominate B"
blocking set accepts "nominate A" OR quorum votes or accepts "nominate A"
node accepts "nominate B"
node accepts "nominate A" node accepts "nominate B"
quorum accepts "nominate A"
quorum accepts "nominate B"
node confirms "nominate A"
node confirms "nominate B"

Balloting Protocol

- A ballot is a pair: <counter,value>, where counter is an integer starting at 1 and value is a candidate from the nomination phase.

- Balloting makes repeated attempts to get the network to reach consensus on *some* candidate in *some* ballot by conducting potentially many federated votes on statements about ballots.

- If ballot <counter,value> appears to be getting stuck, a new vote begins, now on ballot <counter+1,value>.


Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
blocking set accepts "prepare ‹n,C2›"
node accepts "prepare <n,C2›"
blocking set accepts "prepare <1,C›" OR quorum votes or accepts "prepare ‹1,C›"
quorum accepts "prepare ‹n,C2›"
node accepts "prepare <1,C›"
node confirms "prepare ‹n,C2›" node votes "commit <n.C2)"
quorum accepts "prepare <1,C›"
blocking set accepts "commit ‹n,C2›" OR quorum votes or accepts "commit ‹n,C2›"
node confirms "prepare <1,C›" node votes "commit <1,C›"
blocking set accepts "commit ‹n,C2)"
node accepts "commit <n,C2>"
blocking set accepts "commit ‹1,C›" OR quorum votes or accepts "commit ‹1.C›"
node accepts "commit ‹1,C›"
quorum accepts "commit ‹1,C›"
quorum accepts "commit ‹n,C2›"
node confirms "commit <1,C›" node externalizes C
node confirms "commit ‹n,C2›" node externalizes C2

UCDAVIS
COLLEGE OF ENGINEERING

Balloting

- A ballot is a pair: <counter,value>, where counter is an integer starting at 1 and value is a candidate from the nomination phase.
- It could be the node's own candidate, or some peer's candidate that the node comes to accept.
- Balloting makes repeated attempts to get the network to reach consensus on *some* candidate in *some* ballot by conducting potentially many federated votes on statements about ballots.

UCDAVIS COLLEGE OF ENGINEERING

Balloting

So what does “prepared” and “committed” mean?

- A node votes to commit to a ballot when it is convinced that other nodes won’t commit to ballots with different values. Becoming convinced of this is the purpose of the prepare statement.

- The commit vote tells peers that the node will never abort B. In fact, if B is the ballot <N,C>, then “I commit to ballot <N,C>” is implicitly also a vote to *prepare* every ballot from <N,C> through <∞,C>. This extra meaning helps other nodes receiving this message to catch up, if they’re still in earlier phases of the protocol.

Nomination Protocol

Point to note:

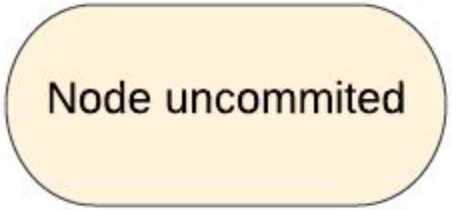
- Not all votes from peers get echoed by a node during nomination, because this could lead to an explosion of different nominees. SCP includes a mechanism for throttling these echoed votes. There is a formula for determining the “priority” of a peer from a node’s point of view, and only high-priority nodes get their votes echoed. The longer nomination takes, the lower the threshold gets, so a node expands the set of peers whose votes it will echo. The priority formula includes the slot number as one of its inputs, so a high-priority peer for one slot may be low-priority for the next, and vice versa.

UCDAVIS COLLEGE OF ENGINEERING

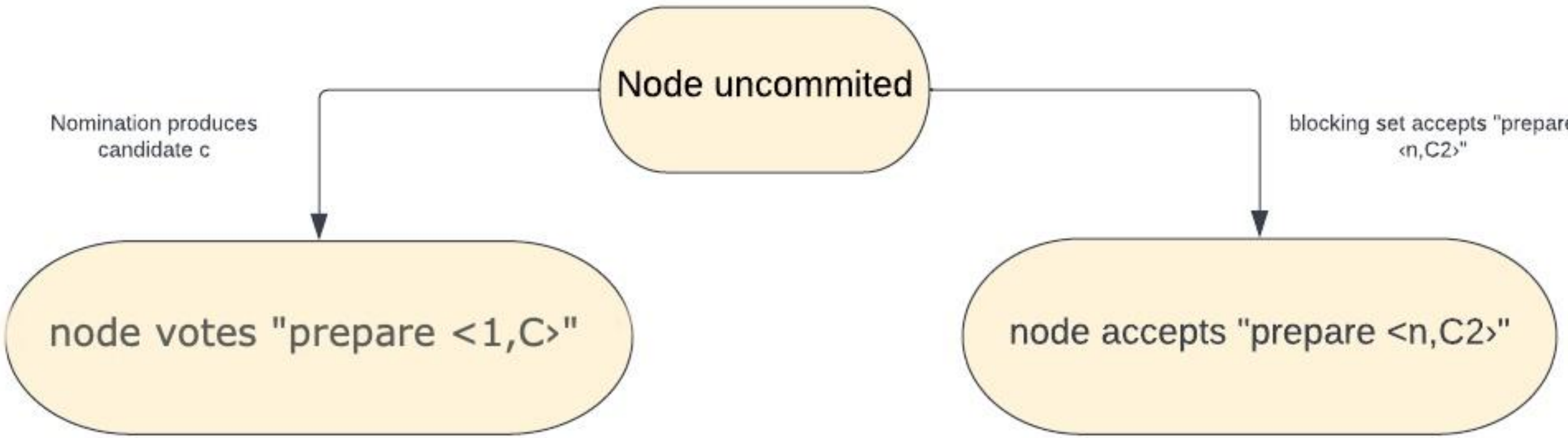
Balloting

Let us look at a visual representation

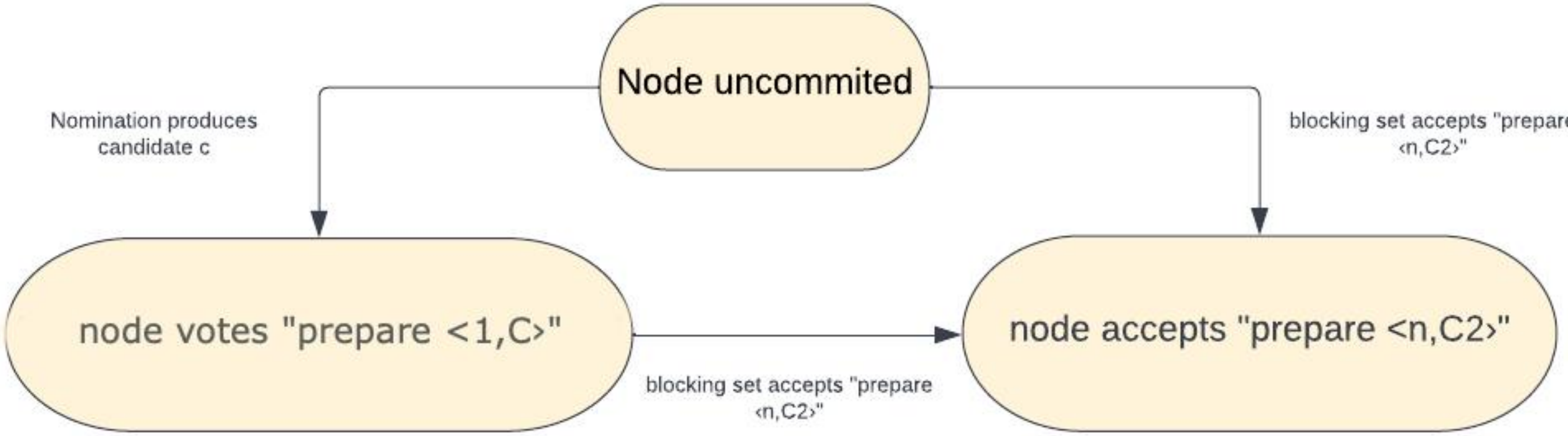
Balloting



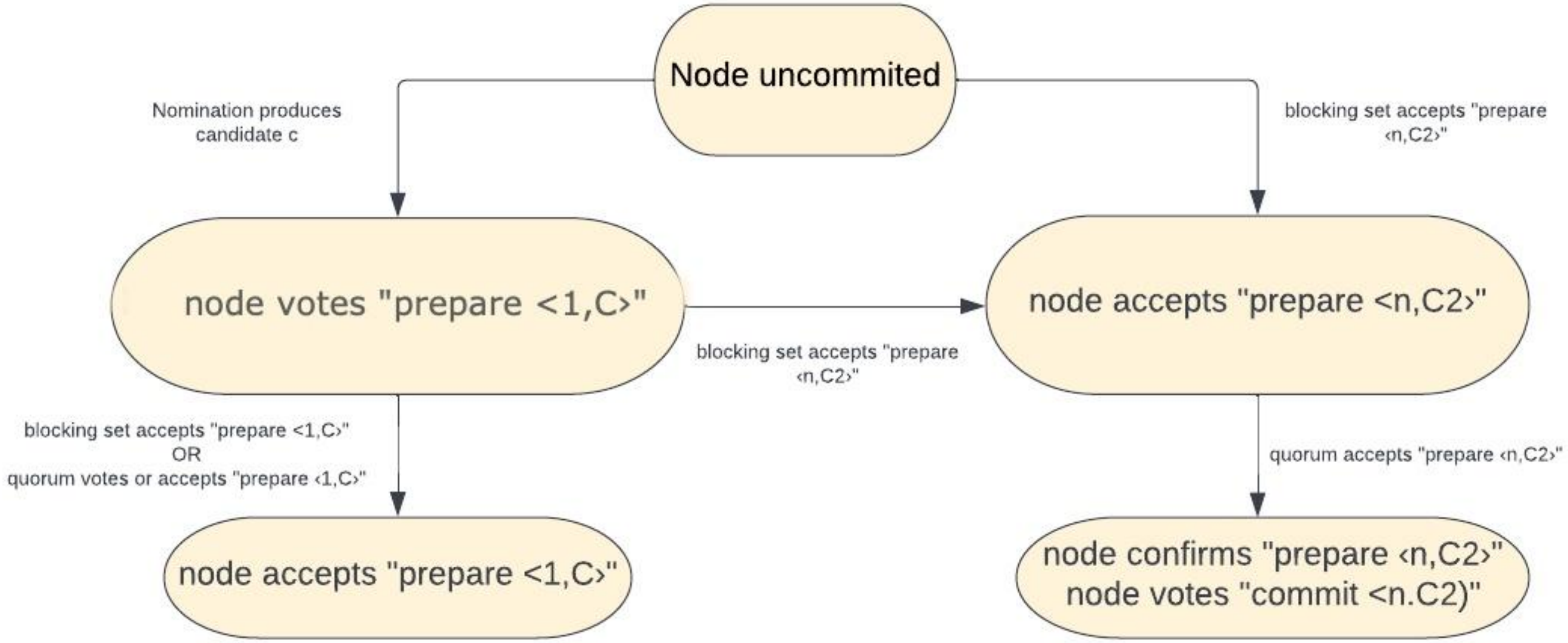
Balloting


Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
node accepts "prepare <n,C2›"

Balloting

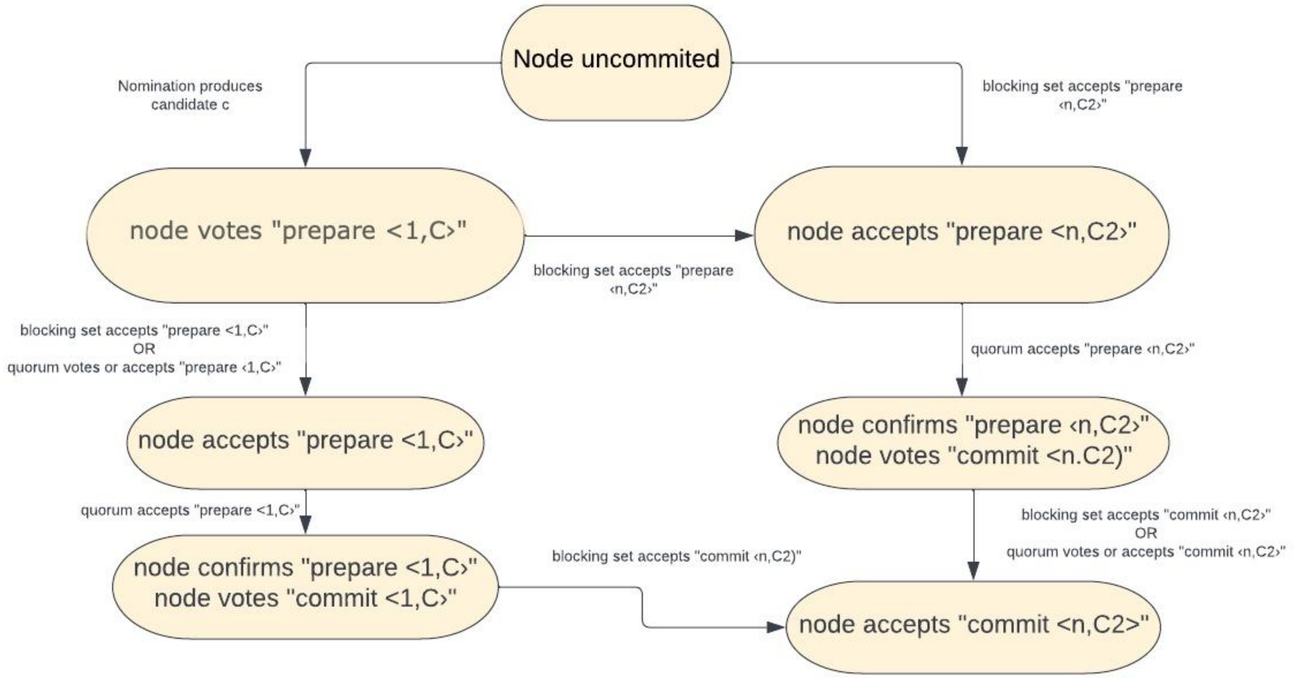

Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
blocking set accepts "prepare ‹n,C2›"
node accepts "prepare <n,C2›"

Balloting

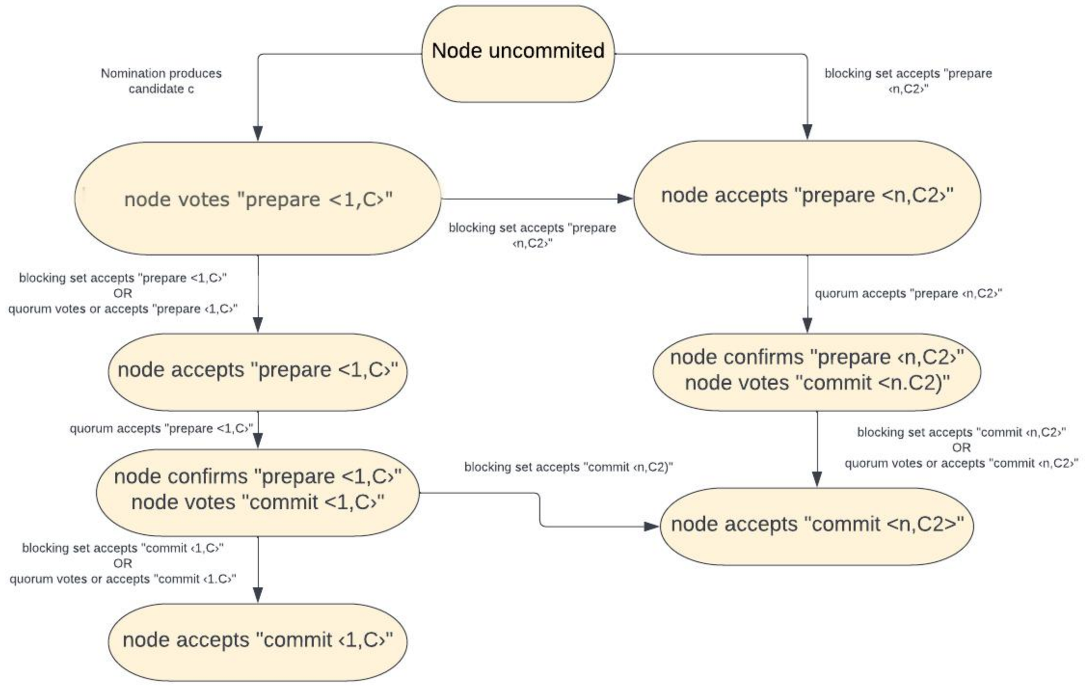

Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
blocking set accepts "prepare ‹n,C2›"
node accepts "prepare <n,C2›"
blocking set accepts "prepare <1,C›" OR quorum votes or accepts "prepare ‹1,C›"
quorum accepts "prepare ‹n,C2›"
node accepts "prepare <1,C›"
node confirms "prepare ‹n,C2›"
node votes "commit <n.C2)"

UCDAVIS COLLEGE OF ENGINEERING

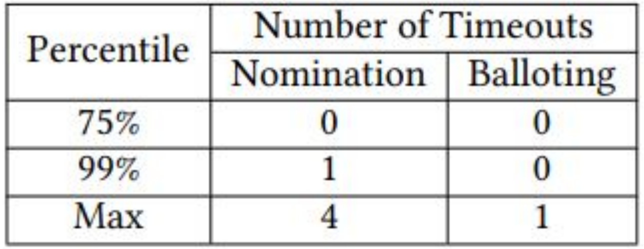
Balloting


Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
blocking set accepts "prepare ‹n,C2›"
node accepts "prepare <n,C2›"
blocking set accepts "prepare <1,C›" OR quorum votes or accepts "prepare ‹1,C›"
quorum accepts "prepare ‹n,C2›"
node accepts "prepare <1,C›"
node confirms "prepare ‹n,C2›" node votes "commit <n.C2)"
quorum accepts "prepare <1,C›"
blocking set accepts "commit ‹n,C2›" OR quorum votes or accepts "commit ‹n,C2›"
node confirms "prepare <1,C›" node votes "commit <1,C›"
blocking set accepts "commit ‹n,C2)"
node accepts "commit <n,C2>"

Balloting

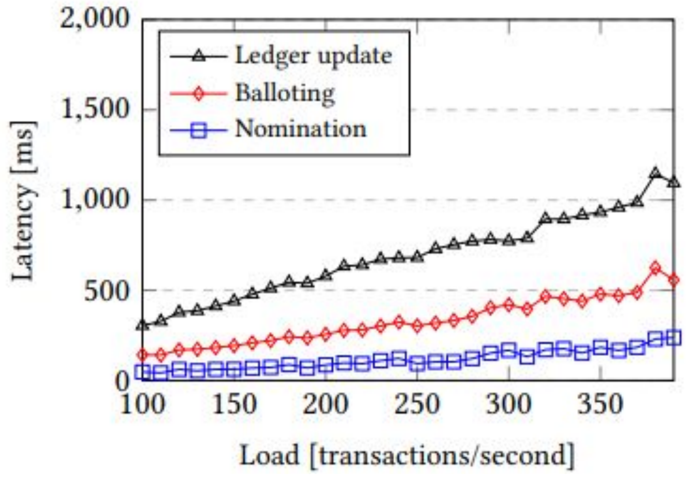

Node uncommited
Nomination produces candidate c
blocking set accepts "prepare ‹n,C2›"
node votes "prepare <1,C›"
blocking set accepts "prepare ‹n,C2›"
node accepts "prepare <n,C2›"
blocking set accepts "prepare <1,C›" OR quorum votes or accepts "prepare ‹1,C›"
quorum accepts "prepare ‹n,C2›"
node accepts "prepare <1,C›"
node confirms "prepare ‹n,C2›" node votes "commit <n.C2)"
quorum accepts "prepare <1,C›"
blocking set accepts "commit ‹n,C2›" OR quorum votes or accepts "commit ‹n,C2›"
node confirms "prepare <1,C›" node votes "commit <1,C›"
blocking set accepts "commit ‹n,C2)"
node accepts "commit <n,C2>"
blocking set accepts "commit ‹1,C›" OR quorum votes or accepts "commit ‹1.C›"
node accepts "commit ‹1,C›"

UCDAVIS COLLEGE OF ENGINEERING

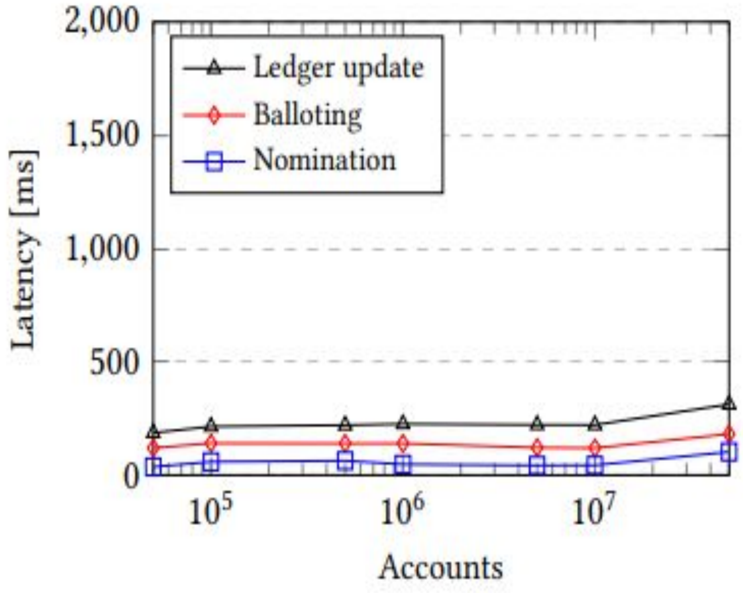


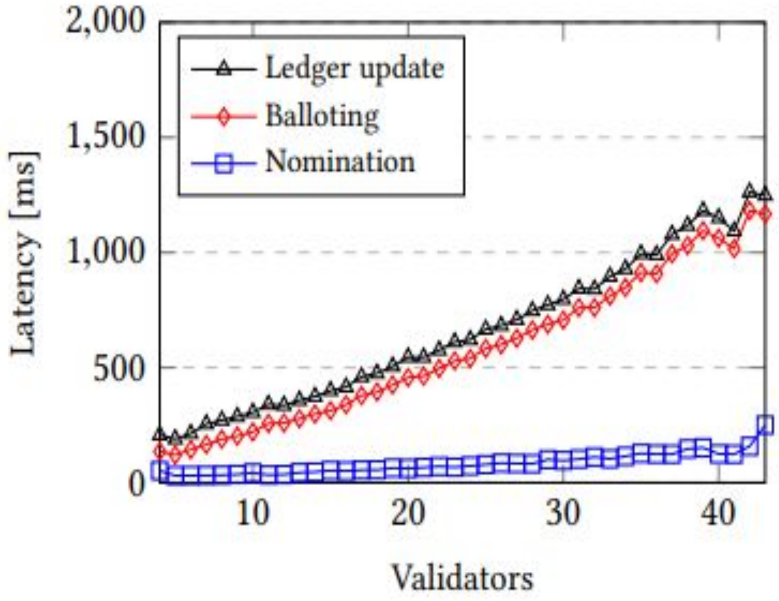
Percentile	Number of Timeouts	
	Nomination	Balloting
75%	0	0
99%	1	0
Max	4	1

Timeouts per ledger over 68 hrs



Latency as transaction load increases





Latency as number of accounts increases

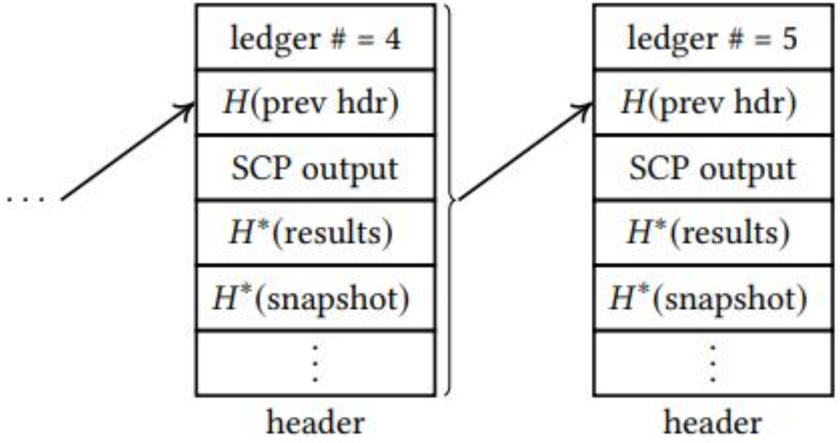
Latency as number of validators increases

Stellar's Payment Network

Ledger Model:

The ledger contents consist a set of ledger entries of four distinct types:

- **Accounts**



Stellar’s Payment Network

Accounts:

- Accounts are the principles that own and issue assets

Each Account consists of:

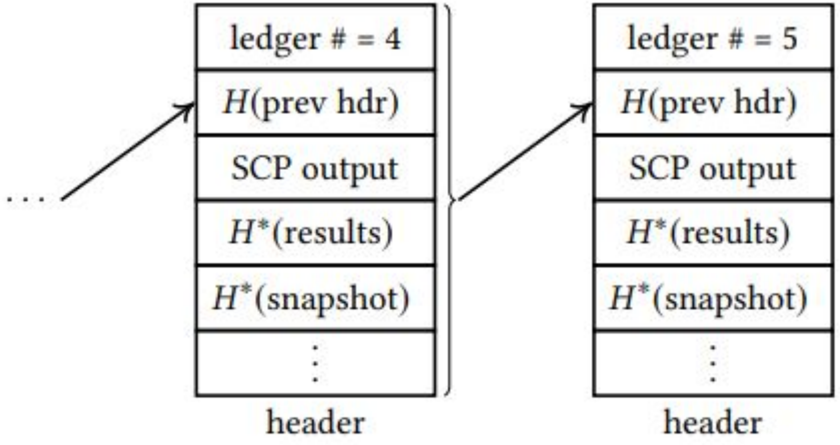
- A sequence number
- Some flags
- A balance in a “native” premined cryptocurrency called XLM
    - This is intended to facilitate market making as a neutral currency

Stellar's Payment Network

Ledger Model:

The ledger contents consist a set of ledger entries of four distinct types:

- **Accounts**
- **Trustlines**



Stellar’s Payment Network

Trustlines:

- Trustlines track the ownership of issued assets
- They are named by a pair consisting of the issuing account and a short asset code (e.g., “USD” or “EUR”)

- Each Trustline specifies:
    - An account
    - An asset
    - The account’s balance in that asset
    - A limit above which the balance cannot rise
    - Some flags

Stellar's Payment Network

Ledger Model:

The ledger contents consist a set of ledger entries of four distinct types:

- **Accounts**
- **Trustlines**
- **Offers**

Stellar’s Payment Network

Ledger Model:

The ledger contents consist a set of ledger entries of four distinct types:

- **Accounts**
- **Trustlines**
- **Offers**
  - Offers correspond to an account’s willingness to trade up to a certain amount of a particular asset for another at a given price on the order book
  - They are automatically matched and filled when buy/sell prices cross

Stellar’s Payment Network

<u>Ledger Model:</u>

The ledger contents consist a set of ledger entries of four distinct types:

- **Accounts**
- **Trustlines**
- **Offers**
- **Account Data**

Stellar's Payment Network

Ledger Model:

The ledger contents consist a set of ledger entries of four distinct types:

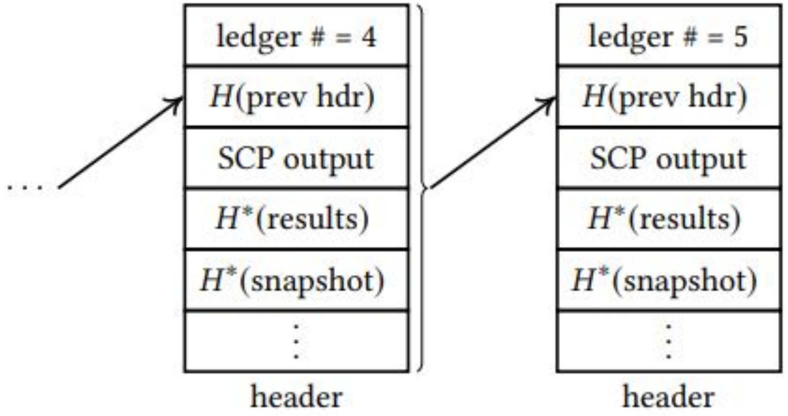
- **Accounts**
- **Trustlines**
- **Offers**
- **Account Data**
    - account data consists of account, key, value triples, allowing account holders to publish small metadata values.

Stellar's Payment Network

Ledger Header:

A ledger header stores global attributes:

- a ledger number
- a hash of the previous ledger header
- the SCP output including a hash of new transactions applied at this ledger
- a hash of the results of those transactions
- a snapshot hash of all ledger entries



Stellar’s Payment Network

Transaction Model:

A transaction consists of a

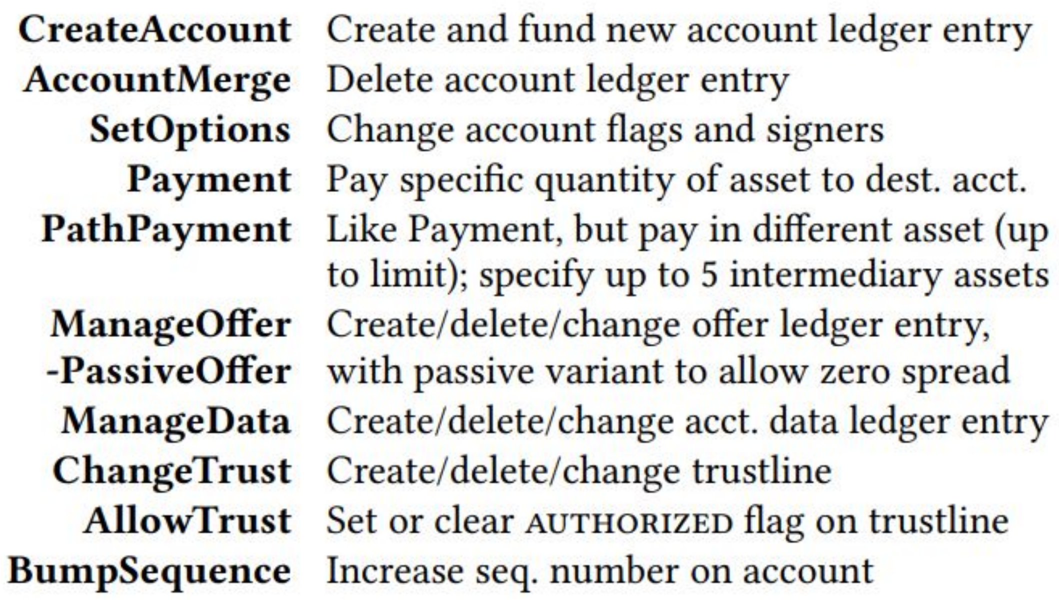
- Source account
- Validity criteria
- A memo
- A list of operations

Stellar’s Payment Network

Transaction Model:

A transaction consists of a

- Source account
- Validity criteria
- A memo
- A list of operations



CreateAccount	Create and fund new account ledger entry
AccountMerge	Delete account ledger entry
SetOptions	Change account flags and signers
Payment	Pay specific quantity of asset to dest. acct.
PathPayment	Like Payment, but pay in different asset (up to limit); specify up to 5 intermediary assets
ManageOffer -PassiveOffer	Create/delete/change offer ledger entry, with passive variant to allow zero spread
ManageData	Create/delete/change acct. data ledger entry
ChangeTrust	Create/delete/change trustline
AllowTrust	Set or clear AUTHORIZED flag on trustline
BumpSequence	Increase seq. number on account

Stellar’s Payment Network

Atomic Transactions:

- if any operation fails, none of them execute. This simplifies multi-way deals.

- **Example:**
  Suppose an issuer creates an asset to represent land deeds
  - User A wants to exchange a small land parcel plus $10,000 for a bigger land parcel owned by B.
  - The two users can both sign a single transaction containing three operations: two land payments and one dollar payment.

Stellar’s Payment Network

Transaction Model - Validity Criteria:

- First Validation Criterion -
- Second Validation Criterion -

Stellar’s Payment Network

Transaction Model - Validity Criteria:

- First Validation Criterion -
  - **Sequence Number**

- Second Validation Criterion -
  -

Stellar’s Payment Network

Transaction Model - Validity Criteria:

- First Validation Criterion -
    - **Sequence Number**

- Second Validation Criterion -
    - **Optional limit on when a transaction can execute**

UCDAVIS COLLEGE OF ENGINEERING

Stellar's Payment Network

Transaction Model - Validity Criteria:

- First Validation Criterion -
    - **Sequence Number**

- Second Validation Criterion -
    - **Optional limit on when a transaction can execute**

What about Transaction Fee??

UCDAVIS
COLLEGE OF ENGINEERING

Stellar's Payment Network

Consensus Values:

For each ledger, Stellar uses SCP to agree on a data structure with three fields:

Transaction Hash (including previous ledger Hash)	A Close Time	Upgrades

UCDAVIS COLLEGE OF ENGINEERING

Stellar’s Payment Network

Upgrades:

- Upgrades adjust global parameters such as the reserve balance, minimum operation fee, and protocol version.
- When combined during nomination, higher fees and protocol version numbers supersede lower ones.
- Upgrades effect governance through a federated-voting

UCDAVIS
COLLEGE OF ENGINEERING

Stellar’s Payment Network

Validators(according to whether its operator wants to participate in governance)

- Governing Validator
  - Desired
  - Valid
  - Invalid
- Non-Governing Validator (default)

Future Work

- Because the performance demands have been so modest to date, Stellar leaves room for many straight-forward optimizations using well-known techniques.

Examples:

- Transactions and SCP messages are broadcast by validators using a naïve flooding protocol, but should ideally use more efficient, structured peer-to-peer multicast.

- Database-heavy ledger update time can be improved through standard batching and prefetching techniques.

UCDAVIS COLLEGE OF ENGINEERING

Thank you

Please feel free to ask any questions