Federated Byzantine Quorum Systems

Álvaro García-Pérez and Alexey Gotsman

IMDEA Software Institute
Blockchains

- Append-only, distributed ledger.
- Uses a Byzantine fault-tolerant (BFT) consensus algorithm to ensure that distributed nodes agree on the next block to append.
Permissioned and permissionless blockchains

- Permissioned blockchains assume a fixed set of participants:
  - classic consensus algorithms, decisions rely on a quorum, i.e., $3f+1$.

- Permissionless blockchains have open membership:
  - often rely on proof-of-work, high energy consumption.
Flexible trust

- Combines quorum systems with decentralisation:
  - The set of participants is fixed, the choice of trust is not.
Flexible trust

• Combines quorum systems with decentralisation:
  • The set of participants is fixed, the choice of trust is not.

• Classic quorum systems:
  • Dissemination quorum systems (DQS) [Malkhi and Reiter, 1998].
  • Allow to choose a tailor-made quorum system.

• Stellar's federated systems [Mazières, 2016]:
  • Federated Byzantine quorum systems (FBQS) [Mazières, 2016].
  • Each participant decides who to trust, and participants may not know the whole system.
## Broadcast and quorum systems

| Classic quorum systems | Stellar's federated systems |
Broadcast and quorum systems

Classic quorum systems

Dissemination quorum systems
[Malkhi and Reiter, 1998]

Bracha broadcast
[Bracha, 1987]

Stellar's federated systems
Broadcast and quorum systems

classic quorum systems

Dissemination quorum systems
[Malkhi and Reiter, 1998]

Bracha broadcast
[Bracha, 1987]

Reliable Byzantine broadcast abstraction

Stellar's federated systems
Broadcast and quorum systems

classic quorum systems

- Dissemination quorum systems [Malkhi and Reiter, 1998]
- Bracha broadcast [Bracha, 1987]
- Reliable Byzantine broadcast abstraction

Stellar's federated systems

- Federated Byzantine quorum systems [Mazières, 2016]
- Stellar broadcast [Mazières, 2016]
Our contribution

classic quorum systems

- Dissemination quorum systems
  [Malkhi and Reiter, 1998]
- Bracha broadcast
  [Bracha, 1987]
- Reliable Byzantine broadcast abstraction

Stellar's federated systems

- Federated Byzantine quorum systems
  [Mazières, 2016]
- Stellar broadcast
  [Mazières, 2016]
- Weakly reliable Byzantine broadcast abstraction

✓ Weakly reliable Byzantine broadcast abstraction
Dissemination Quroum System (DQS)
\( V = \{1,2,3,4\} \)

\((\mathbb{Q} : 2^4, \mathbb{B} : 2^4)\)

\( U_1 = \{1,2\} \in \mathbb{Q} \)
\( U_2 = \{1,3,4\} \in \mathbb{Q} \)
\( U_3 = \{1,2,3\} \in \mathbb{Q} \)
\( U_4 = \{1,2,3,4\} \in \mathbb{Q} \)

\( B_1 = \{2\} \in \mathbb{B} \)
\( B_2 = \{3,4\} \in \mathbb{B} \)

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathbb{Q} \) cannot lie within any element \( B \) of \( \mathbb{B} \).

- **(Availability)** For any element \( B \) of \( \mathbb{B} \) there exists some quorum \( U \) in \( \mathbb{Q} \) that has empty intersection with \( B \).
\( \mathcal{V} = \{1, 2, 3, 4\} \)

\( (\mathcal{Q} : 2^\mathcal{V}, \mathbb{B} : 2^\mathcal{V}) \)

\( U_1 = \{1, 2\} \in \mathcal{Q} \)
\( U_2 = \{1, 3, 4\} \in \mathcal{Q} \)
\( U_3 = \{1, 2, 3\} \in \mathcal{Q} \)
\( U_4 = \{1, 2, 3, 4\} \in \mathcal{Q} \)

\( B_1 = \{2\} \in \mathbb{B} \)
\( B_2 = \{3, 4\} \in \mathbb{B} \)

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathcal{Q} \) cannot lie within any element \( B \) of \( \mathbb{B} \).

- **(Availability)** For any element \( B \) of \( \mathbb{B} \) there exists some quorum \( U \) in \( \mathcal{Q} \) that has empty intersection with \( B \).
\( V = \{1,2,3,4\} \)

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathcal{Q} \) cannot lie within any element \( B \) of \( \mathcal{B} \).

- **(Availability)** For any element \( B \) of \( \mathcal{B} \) there exists some quorum \( U \) in \( \mathcal{Q} \) that has empty intersection with \( B \).
\[ V = \{1,2,3,4\} \]

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathbb{Q} \) cannot lie within any element \( B \) of \( \mathbb{B} \).

- **(Availability)** For any element \( B \) of \( \mathbb{B} \) there exists some quorum \( U \) in \( \mathbb{Q} \) that has empty intersection with \( B \).
\( \mathbb{V} = \{1, 2, 3, 4\} \)

\((Q : 2^{2^V}, \mathbb{B} : 2^{2^V})\)

\(U_1 = \{1, 2\} \in Q\)
\(U_2 = \{1, 3, 4\} \in Q\)
\(U_3 = \{1, 2, 3\} \in Q\)
\(U_4 = \{1, 2, 3, 4\} \in Q\)

\(B_1 = \{2\} \in \mathbb{B}\)
\(B_2 = \{3, 4\} \in \mathbb{B}\)

- **(Consistency)** The intersection of any two quorums \(U\) and \(U'\) in \(Q\) cannot lie within any element \(B\) of \(\mathbb{B}\).
- **(Availability)** For any element \(B\) of \(\mathbb{B}\) there exists some quorum \(U\) in \(Q\) that has empty intersection with \(B\).
\[ V = \{1, 2, 3, 4\} \]

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathcal{Q} \) cannot lie within any element \( B \) of \( \mathcal{B} \).

- **(Availability)** For any element \( B \) of \( \mathcal{B} \) there exists some quorum \( U \) in \( \mathcal{Q} \) that has empty intersection with \( B \).
\( V = \{1, 2, 3, 4\} \)

\[ ( \mathcal{Q} : 2^{2^V}, \mathcal{B} : 2^{2^V} ) \]

\[ U_1 = \{1, 2\} \in \mathcal{Q} \]
\[ U_2 = \{1, 3, 4\} \in \mathcal{Q} \]
\[ U_3 = \{1, 2, 3\} \in \mathcal{Q} \]
\[ U_4 = \{1, 2, 3, 4\} \in \mathcal{Q} \]

\[ B_1 = \{2\} \in \mathcal{B} \]
\[ B_2 = \{3, 4\} \in \mathcal{B} \]

- **(Consistency)** The intersection of any two quorums \( U \) and \( U' \) in \( \mathcal{Q} \) cannot lie within any element \( B \) of \( \mathcal{B} \).

- **(Availability)** For any element \( B \) of \( \mathcal{B} \) there exists some quorum \( U \) in \( \mathcal{Q} \) that has empty intersection with \( B \).
DQS and threshold models

- DQS generalises usual BFT models with threshold $f$ and $n = 3f+1$ servers.
DQS and threshold models

- DQS generalises usual BFT models with threshold $f$ and $n = 3f+1$ servers.

$f = 1, \ n = 4$

1 2 3 4
DQS generalises usual BFT models with threshold $f$ and $n = 3f + 1$ servers.

$$f = 1, \ n = 4$$

Quorums equal or bigger than $2f + 1 = 3$

$$\mathcal{Q} = \{ \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\} \}$$

Fail-prone sets exactly $f = 1$

$$\mathcal{B} = \{ \{1\}, \{2\}, \{3\}, \{4\} \}$$
DQS and threshold models

- DQS generalises usual BFT models with threshold $f$ and $n = 3f+1$ servers.

$$f = 1, \ n = 4$$

Quorums equal or bigger than $2f+1 = 3$

$\mathcal{Q} = \{ \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\} \}$

Fail-prone sets exactly $f = 1$

$\mathcal{B} = \{ \{1\}, \{2\}, \{3\}, \{4\} \}$

- (Consistency) Every two quorums intersect in at least $f+1$ servers.
- (Availability) If $f$ servers fail, the remaining ones constitutes a quorum.
Bracha Broadcast
Example: $3f+1$
Example: $3f+1$
Example: $3f+1$

To broadcast a value $a$, the client sends $[\text{BCAST } a]$ to every server.
After receiving [BCAST a], a server sends [ECHO a] to every server.
Example: $3f+1$

<table>
<thead>
<tr>
<th>Client</th>
<th>Database 1</th>
<th>Database 2</th>
<th>Database 3</th>
<th>Database 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>receive \texttt{BCAST a}</td>
<td>receive \texttt{BCAST a}</td>
<td>receive \texttt{BCAST b}</td>
<td></td>
</tr>
<tr>
<td>send \texttt{ECHO a} to all</td>
<td>send \texttt{ECHO a} to all</td>
<td>send \texttt{ECHO a} to 1,2</td>
<td>send \texttt{ECHO b} to all</td>
<td></td>
</tr>
</tbody>
</table>
After receiving \([\text{ECHO } a]\) from a quorum, a server sends \([\text{READY } a]\) to every server.
Example: $3f+1$

<table>
<thead>
<tr>
<th>Client</th>
<th>Server 1</th>
<th>Server 2</th>
<th>Server 3</th>
<th>Server 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>receive [BCAST $a$]</td>
<td>receive [BCAST $a$]</td>
<td>receive [BCAST $b$]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>send [ECHO $a$] to all</td>
<td>send [ECHO $a$] to all</td>
<td>send [ECHO $a$] to 1,2</td>
<td>send [ECHO $b$] to all</td>
<td></td>
</tr>
<tr>
<td>send [READY $a$] to all</td>
<td>send [READY $a$] to all</td>
<td></td>
<td>send [READY $a$] to all</td>
<td></td>
</tr>
</tbody>
</table>

After receiving [READY $a$] from a set $B$ such that $\forall B' \in \mathbb{B}, B \not\subseteq B'$, a server sends [READY $a$] to every server.
Example: $3f+1$

After receiving \texttt{[READY a]} from a quorum, a server delivers value $a$. 

<table>
<thead>
<tr>
<th>receive [BCAST a]</th>
<th>send [ECHO a] to all</th>
<th>send [ECHO a] to 1,2</th>
<th>send [ECHO b] to all</th>
</tr>
</thead>
<tbody>
<tr>
<td>send [READY a] to all</td>
<td>send [READY a] to all</td>
<td>send [READY a] to all</td>
<td>send [READY a] to all</td>
</tr>
<tr>
<td>deliver(a)</td>
<td>deliver(a)</td>
<td></td>
<td>deliver(a)</td>
</tr>
</tbody>
</table>
Bracha broadcast satisfies the specification of reliable Byzantine broadcast when all faulty servers belong to some element of $\mathbb{B}$:

- **Safety:** If some correct server delivers a value $a$ and another correct server delivers a value $b$, then $a = b$.
- **Liveness:** If a correct server delivers a value, then every correct server eventually delivers a value.
Bracha broadcast satisfies the specification of reliable Byzantine broadcast when all faulty servers belong to some element of $\mathbb{B}$:

- **Safety**: If some correct server delivers a value $a$ and another correct server delivers a value $b$, then $a = b$.

- **Liveness**: If a correct server delivers a value, then every correct server eventually delivers a value.

The protocol needs to compute $\mathbb{B}$, which requires global information!
Federated Byzantine Quroum Systems (FBQS)
$\mathbb{V} = \{1, 2, 3, 4\}$

$\mathbb{S} : \mathbb{V} \rightarrow 2^{\mathbb{V}}$

$\mathbb{S}(1) = \{\{1, 2\}, \{1, 4\}\}$
$\mathbb{S}(2) = \{\{1, 2\}\}$
$\mathbb{S}(3) = \{\{1, 3\}\}$
$\mathbb{S}(4) = \{\{3, 4\}\}$
$V = \{1,2,3,4\}$

$\mathbb{S} : V \rightarrow 2^V$

$\mathbb{S}(1) = \{\{1,2\},\{1,4\}\}$
$\mathbb{S}(2) = \{\{1,2\}\}$
$\mathbb{S}(3) = \{\{1,3\}\}$
$\mathbb{S}(4) = \{\{3,4\}\}$
\( V = \{1, 2, 3, 4\} \)

\( S : V \rightarrow 2^V \)

\( S(1) = \{\{1, 2\}, \{1, 4\}\} \)

\( S(2) = \{\{1, 2\}\} \)

\( S(3) = \{\{1, 3\}\} \)

\( S(4) = \{\{3, 4\}\} \)
FBQS

\[ \mathbb{V} = \{1, 2, 3, 4\} \]

\[ S : \mathbb{V} \rightarrow 2^{\mathbb{V}} \]

\[ S(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S(2) = \{\{1, 2\}\} \]
\[ S(3) = \{\{1, 3\}\} \]
\[ S(4) = \{\{3, 4\}\} \]
$S : V \rightarrow 2^V$

$S(1) = \{\{1,2\},\{1,4\}\}$
$S(2) = \{\{1,2\}\}$
$S(3) = \{\{1,3\}\}$
$S(4) = \{\{3,4\}\}$

$V = \{1,2,3,4\}$
$\mathbb{V} = \{1, 2, 3, 4\}$

$\mathbb{S} : \mathbb{V} \rightarrow 2^\mathbb{V}$

$\mathbb{S}(1) = \{\{1, 2\}, \{1, 4\}\}$
$\mathbb{S}(2) = \{\{1, 2\}\}$
$\mathbb{S}(3) = \{\{1, 3\}\}$
$\mathbb{S}(4) = \{\{3, 4\}\}$
$S : V \rightarrow 2^V$

$S(1) = \{\{1, 2\}, \{1, 4\}\}$
$S(2) = \{\{1, 2\}\}$
$S(3) = \{\{1, 3\}\}$
$S(4) = \{\{3, 4\}\}$

$V = \{1, 2, 3, 4\}$

$U_1 = \{1, 2\} \in \mathbb{Q}$
$S : V \rightarrow 2^V$

$S(1) = \{\{1, 2\}, \{1, 4\}\}$
$S(2) = \{\{1, 2\}\}$
$S(3) = \{\{1, 3\}\}$
$S(4) = \{\{3, 4\}\}$

$V = \{1, 2, 3, 4\}$

$U_1 = \{1, 2\} \in \mathbb{Q}$
$U_2 = \{1, 3, 4\} \in \mathbb{Q}$
$S : V \rightarrow 2^V$

$S(1) = \{1, 2\}, \{1, 4\}$
$S(2) = \{1, 2\}$
$S(3) = \{1, 3\}$
$S(4) = \{3, 4\}$

$V = \{1, 2, 3, 4\}$

$U_1 = \{1, 2\} \in \mathbb{Q}$
$U_2 = \{1, 3, 4\} \in \mathbb{Q}$
$U_3 = \{1, 2, 3\} \in \mathbb{Q}$
$V = \{1, 2, 3, 4\}$

$S : V \rightarrow 2^{2V}$

$S(1) = \{\{1, 2\}, \{1, 4\}\}$
$S(2) = \{\{1, 2\}\}$
$S(3) = \{\{1, 3\}\}$
$S(4) = \{\{3, 4\}\}$

$U_1 = \{1, 2\} \in \mathbb{Q}$
$U_2 = \{1, 3, 4\} \in \mathbb{Q}$
$U_3 = \{1, 2, 3\} \in \mathbb{Q}$
$U_4 = \{1, 2, 3, 4\} \in \mathbb{Q}$
Given a set of faulty servers, $\mathbb{V}_{\text{int}}$ is the biggest quorum $\mathbb{V}_{\text{int}} \in \mathbb{Q}$ such that:

- $\forall v \in \mathbb{V}_{\text{int}}, v$ is correct,
- $\mathbb{Q}|_{\mathbb{V}_{\text{int}}}$ has quorum intersection.
Given a set of faulty servers, $\mathbb{V}_{\text{int}}$ is the biggest quorum $\mathbb{V}_{\text{int}} \in \mathbb{Q}$ such that:

- $\forall v \in \mathbb{V}_{\text{int}}, v$ is correct,
- $\mathbb{Q}|_{\mathbb{V}_{\text{int}}}$ has quorum intersection.
Given a set of faulty servers, \( V_{\text{int}} \) is the biggest quorum \( V_{\text{int}} \in \mathbb{Q} \) such that:

- \( \forall v \in V_{\text{int}}, v \) is correct,
- \( \mathbb{Q}|V_{\text{int}} \) has quorum intersection.
Given a set of faulty servers, $\mathbb{V}_{\text{int}}$ is the biggest quorum $\mathbb{V}_{\text{int}} \in \mathbb{Q}$ such that:

- $\forall v \in \mathbb{V}_{\text{int}}, v$ is correct,
- $\mathbb{Q}|\mathbb{V}_{\text{int}}$ has quorum intersection.
Given a set of faulty servers, \( \mathbb{V}_{\text{int}} \) is the biggest quorum \( \mathbb{V}_{\text{int}} \in \mathcal{Q} \) such that:

- \( \forall v \in \mathbb{V}_{\text{int}}, v \) is correct,
- \( \mathcal{Q}|_{\mathbb{V}_{\text{int}}} \) has quorum intersection.
In threshold models like $3f+1$, the notions of intact and correct coincide.

Given a set of faulty servers, $\mathbb{V}_{\text{int}}$ is the biggest quorum $\mathbb{V}_{\text{int}} \in \mathbb{Q}$ such that:

- $\forall v \in \mathbb{V}_{\text{int}}, v$ is correct,
- $\mathbb{Q}|\mathbb{V}_{\text{int}}$ has quorum intersection.
Mapping FBQS into DQS

\[
\begin{align*}
S &: V \rightarrow 2^V \\
S(1) &= \{\{1,2\}, \{1,4\}\} \\
S(2) &= \{\{1,2\}\} \\
S(3) &= \{\{1,3\}\} \\
S(4) &= \{\{3,4\}\}
\end{align*}
\]

\[
\begin{align*}
U_1 &= \{1,2\} \in Q \\
U_2 &= \{1,3,4\} \in Q \\
U_3 &= \{1,2,3\} \in Q \\
U_4 &= \{1,2,3,4\} \in Q
\end{align*}
\]

\[
\begin{align*}
B_1 &= \{2\} \in B \\
B_2 &= \{3,4\} \in B
\end{align*}
\]
Mapping FBQS into DQS

$S : V \rightarrow 2^V$

$S(1) = \{\{1,2\}, \{1,4\}\}$
$S(2) = \{\{1,2\}\}$
$S(3) = \{\{1,3\}\}$
$S(4) = \{\{3,4\}\}$

The elements in $\mathbb{B}$ are the maximal sets whose failure leave some intact server in the system.

$U_1 = \{1,2\} \in \mathbb{Q}$
$U_2 = \{1,3,4\} \in \mathbb{Q}$
$U_3 = \{1,2,3\} \in \mathbb{Q}$
$U_4 = \{1,2,3,4\} \in \mathbb{Q}$

$B_1 = \{2\} \in \mathbb{B}$
$B_2 = \{3,4\} \in \mathbb{B}$
Stellar Broadcast
\( \nu \)-blocking mechanism

\( \mathbb{S} : \mathbb{V} \rightarrow 2^\mathbb{V} \)

\( \mathbb{S}(1) = \{\{1, 2\}, \{1, 4\}\} \)

\( \mathbb{S}(2) = \{\{1, 2\}\} \)

\( \mathbb{S}(3) = \{\{1, 3\}\} \)

\( \mathbb{S}(4) = \{\{3, 4\}\} \)

\( \mathbb{V} = \{1, 2, 3, 4\} \)

\( B_1 \supseteq \{2, 4\} \) is 1-blocking
$v$-blocking mechanism

$\mathbb{S} : V \rightarrow 2^V$

$\mathbb{S}(1) = \{\{1, 2\}, \{1, 4\}\}$
$\mathbb{S}(2) = \{\{1, 2\}\}$
$\mathbb{S}(3) = \{\{1, 3\}\}$
$\mathbb{S}(4) = \{\{3, 4\}\}$

$V = \{1, 2, 3, 4\}$

$B_1 \supseteq \{2, 4\}$ is 1-blocking

If $v$ is intact, only intact servers can block $v$. 
**v-blocking mechanism**

\[ \mathbb{V} = \{1, 2, 3, 4\} \]

\[
\begin{align*}
\mathbb{S}(1) &= \{\{1, 2\}, \{1, 4\}\} \\
\mathbb{S}(2) &= \{\{1, 2\}\} \\
\mathbb{S}(3) &= \{\{1, 3\}\} \\
\mathbb{S}(4) &= \{\{3, 4\}\}
\end{align*}
\]

If \( v \) is intact, only intact servers can block \( v \).

A \( v \)-blocking set can be computed by \( v \) locally!

\( B_1 \supseteq \{2, 4\} \) is 1-blocking.
Example:
Example:
Example:

To broadcast a value $a$, the client sends $\texttt{[BCAST a]}$ to every server.
Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>receive</td>
<td>[BCAST a]</td>
<td>[BCAST a]</td>
<td></td>
<td>[BCAST b]</td>
</tr>
<tr>
<td>send</td>
<td>[ECHO a] to all</td>
<td>[ECHO a] to all</td>
<td></td>
<td>[ECHO b] to all</td>
</tr>
</tbody>
</table>

After receiving [BCAST a], a server sends [ECHO a] to every server.
Example:

receive [BCAST a]  receive [BCAST a]  receive [BCAST b]  
send [ECHO a] to all send [ECHO a] to all send [READY b] to 4 send [ECHO b] to all
Example:

After receiving [ECHO a] from a quorum, a server sends [READY a] to every server.
Example:

After receiving `[READY \(a\)]` from a \(v\)-blocking set, \(v\) sends `[READY \(a\)]` to every server.
Example:

After receiving \([\text{READY } a]\) from a quorum, a server delivers value \(a\).
Example:

After receiving \([\text{READY } a]\) from a quorum, a server delivers value \(a\).
Stellar broadcast satisfies the specification of weakly reliable Byzantine broadcast when the faulty servers leave at least one intact server:

- **Safety:** If some correct server delivers a value $a$ and another correct server delivers a value $b$, then $a = b$.
- **Liveness:** If a correct server delivers a value, then every intact server eventually delivers a value.
Weakly reliable Byzantine broadcast

Stellar broadcast satisfies the specification of weakly reliable Byzantine broadcast when the faulty servers leave at least one intact server:

- **Safety**: If some correct server delivers a value \(a\) and another correct server delivers a value \(b\), then \(a = b\).

- **Liveness**: If a correct server delivers a value, then every intact server eventually delivers a value.
Stellar broadcast satisfies the specification of weakly reliable Byzantine broadcast when the faulty servers leave at least one intact server:

- **Safety**: If some correct server delivers a value $a$ and another correct server delivers a value $b$, then $a = b$.

- **Liveness**: If a correct server delivers a value, then every intact server eventually delivers a value.

**Trade-off**: operating on local information weakens the liveness properties to intact servers.
Subjective FBQS
Subjective FBQS

\[ V = \{1, 2, 3, 4\} \]

\[ S : V \to 2^V \setminus \{\emptyset\} \]

\[ S(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S(2) = \{\{1, 2\}\} \]
\[ S(3) = \{\{1, 3\}\} \]
\[ S(4) = \{\{3, 4\}\} \]
Subjective FBQS

\[ \mathbb{V} = \{1, 2, 3, 4\} \]
Subjective FBQS

$V = \{1, 2, 3, 4\}$

$S_1 = S_4$
Subjective FBQS

\[ V = \{1, 2, 3, 4\} \]

\[ S_1 = S_4 \]
Subjective FBQS

\[ \mathcal{V} = \{1, 2, 3, 4\} \]

$S_1 = S_4$

$S_2$
Subjective FBQS

\[ V = \{1, 2, 3, 4\} \]

\[ \mathbb{S}_1(1) = \mathbb{S}_4(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ \mathbb{S}_1(2) = \mathbb{S}_4(2) = \{\{1, 2\}\} \]
\[ \mathbb{S}_1(3) = \mathbb{S}_4(3) = \{\{1, 3\}\} \]
\[ \mathbb{S}_1(4) = \mathbb{S}_4(4) = \{\{3, 4\}\} \]

\[ \mathbb{S}_2(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ \mathbb{S}_2(2) = \{\{1, 2\}\} \]
\[ \mathbb{S}_2(3) = \{\{2, 3\}\} \]
\[ \mathbb{S}_2(4) = \{\{3, 4\}\} \]
Subjective FBQS

$V = \{1, 2, 3, 4\}$

$S_1 = S_4$

$S_1(1) = S_4(1) = \{\{1, 2\}, \{1, 4\}\}$
$S_1(2) = S_4(2) = \{\{1, 2\}\}$
$S_1(3) = S_4(3) = \{\{1, 3\}\}$
$S_1(4) = S_4(4) = \{\{3, 4\}\}$

$S_2$

$S_2(1) = \{\{1, 2\}, \{1, 4\}\}$
$S_2(2) = \{\{1, 2\}\}$
$S_2(3) = \{\{2, 3\}\}$
$S_2(4) = \{\{3, 4\}\}$
Subjective FBQS

\[ \mathcal{V} = \{1, 2, 3, 4\} \]

\[ U_2 = \{1, 3, 4\} \text{ is not a quorum in } \mathcal{S}_2! \]

\[ S_1(1) = S_4(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S_1(2) = S_4(2) = \{\{1, 2\}\} \]
\[ S_1(3) = S_4(3) = \{\{1, 3\}\} \]
\[ S_1(4) = S_4(4) = \{\{3, 4\}\} \]

\[ S_2(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S_2(2) = \{\{1, 2\}\} \]
\[ S_2(3) = \{\{2, 3\}\} \]
\[ S_2(4) = \{\{3, 4\}\} \]
Subjective FBQS

$V = \{1, 2, 3, 4\}$

$U_2 = \{1, 3, 4\}$ is not a quorum in $S_2$!

$S_1(1) = S_4(1) = \{\{1, 2\}, \{1, 4\}\}$

$S_1(2) = S_4(2) = \{\{1, 2\}\}$

$S_1(3) = S_4(3) = \{\{1, 3\}\}$

$S_1(4) = S_4(4) = \{\{3, 4\}\}$

$S_2(1) = \{\{1, 2\}, \{1, 4\}\}$

$S_2(2) = \{\{1, 2\}\}$

$S_2(3) = \{\{2, 3\}\}$

$S_2(4) = \{\{3, 4\}\}$
Subjective FBQS

\[ V = \{1, 2, 3, 4\} \]

\[ U_2 = \{1, 3, 4\} \] is not a quorum in \( S_2 \)!

\[ S_1(1) = S_4(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S_1(2) = S_4(2) = \{\{1, 2\}\} \]
\[ S_1(3) = S_4(3) = \{\{1, 3\}\} \]
\[ S_1(4) = S_4(4) = \{\{3, 4\}\} \]

\[ S_2(1) = \{\{1, 2\}, \{1, 4\}\} \]
\[ S_2(2) = \{\{1, 2\}\} \]
\[ S_2(3) = \{\{2, 3\}\} \]
\[ S_2(4) = \{\{3, 4\}\} \]
Subjective FBQS

$\mathbb{S}_1 = \mathbb{S}_4$

$\mathbb{S}_1(1) = \mathbb{S}_4(1) = \{\{1,2\}, \{1,4\}\}$
$\mathbb{S}_1(2) = \mathbb{S}_4(2) = \{\{1,2\}\}$
$\mathbb{S}_1(3) = \mathbb{S}_4(3) = \{\{1,3\}\}$
$\mathbb{S}_1(4) = \mathbb{S}_4(4) = \{\{3,4\}\}$

$\mathbb{S}_2(1) = \{\{1,2\}, \{1,4\}\}$
$\mathbb{S}_2(2) = \{\{1,2\}\}$
$\mathbb{S}_2(3) = \{\{2,3\}\}$
$\mathbb{S}_2(4) = \{\{3,4\}\}$

$\mathbb{V}_{\text{int}} = \{1,2\}$

$U_2 = \{1,3,4\}$ is not a quorum in $\mathbb{S}_2$!
Subjective FBQS

\[ S_1 = S_4 \]

\[ V_{\text{int}} = \{1,2\} \]

\[ S_1(1) = S_4(1) = \{1,3\} \]
\[ S_1(2) = S_4(2) = \{1,2\} \]
\[ S_1(3) = S_4(3) = \{1,3\} \]
\[ S_1(4) = S_4(4) = \{3,4\} \]

\[ U_2 = \{1,3,4\} \text{ is not a quorum in } S_2! \]

Stellar broadcast over a subjective FBQS with some intact server implements weak reliable Byzantine broadcast.
Work in progress

- Proof of correctness of the whole Stellar consensus protocol.
- Relation between Stellar consensus and existing BFT algorithms.
Conclusions

- An FBQS maps into a DQS, so off-the-shelf DQS algorithms can be run over FBQS:
  - Trade-off between servers relying on global/local information and liveness properties for correct/intact servers.

- If the set of intact servers coincides with the set of correct servers, then Stellar broadcast and Bracha broadcast are observationally equivalent.

- We prove Stellar broadcast correct when servers lie about their slices.
Conclusions

• An FBQS maps into a DQS, so off-the-shelf DQS algorithms can be run over FBQS:
  • Trade-off between servers relying on global/local information and liveness properties for correct/intact servers.

• If the set of intact servers coincides with the set of correct servers, then Stellar broadcast and Bracha broadcast are observationally equivalent.

• We prove Stellar broadcast correct when servers lie about their slices.

Thanks!