An Economic Model of Consensus on Distributed Ledgers

Hanna Halaburda, NYU Stern

Zhiguo He, Chicago Booth and NBER

Jiasun Li, George Mason

Byzantine Fault Tolerance (BFT) in Blockchains

ignite

BFT is Older than Blockchains

- Classic problem with known solutions in distributed databases
  - going back to late 1970's
- Resurgence of interest
  - blockchains – distributed ledgers
  - BFT protocols as guidance for designing blockchain protocols

- Crucial difference in adversarial environment
  - traditional distributed databases:
    - some nodes may fail or be hacked, others follow the protocol
  - blockchains:
    - nodes are independent entities, individually payoff-maximizing
    - **every node** decides whether it's worth for them to follow the protocol or deviate
  - need for economic incentives in analysis of BFT consensus

Economic Model of BFT Consensus

- Characterize equilibria
  - not every design achieves consensus in presence of rational agents
  - designs differ in how costly they are
    - incentives → cost of the system

- Show how the **design** of the protocol affects the **system cost** of incentives needed for consensus **in equilibrium**
- One example:
  - traditional (and current blockchain) BFT protocols recommend that nodes send and forward messages as much as they can
  - we show that in the presence of message loss, it may be prohibitively costly to achieve reliable consensus with such strategies
  - lowering the probability with which the message is sent achieves consensus at a lower system cost

Byzantine Fault Tolerant (BFT) protocols

Classic problem in computer science (eg, Lamport, Shostak, Pease `82)

- Distributed computer nodes communicate with each other to …
- Reach consensus based on “local” information (no “global” knowledge)
- Byzantine nodes behave arbitrarily
- Stipulate “honest” strategies for non-Byzantine nodes
- Widely used in tech companies to maintain distributed databases

Byzantine Fault Tolerant (BFT) protocols

Classic problem in computer science (since Lamport, Shostak, Pease `82)

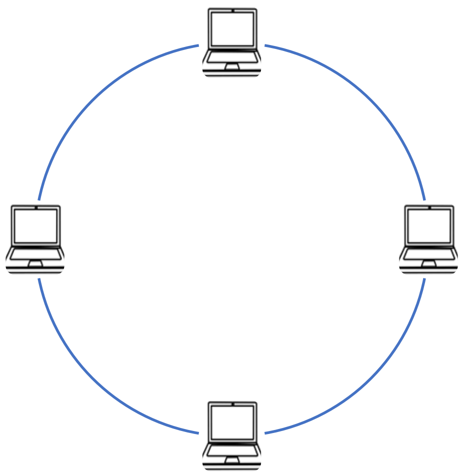
- Distributed computer nodes communicate with each other to …
- Reach consensus based on “local” information (no “global” knowledge)
- Byzantine nodes behave arbitrarily
- ~~Stipulate “honest” strategies for non-Byzantine nodes~~
- Widely used in tech companies to maintain distributed databases

This paper (given that blockchains live in trustless environments)

- Non-Byzantine nodes are **rational**
- **Ambiguity averse** (Knightian uncertain) about Byzantine strategies

Consensus game

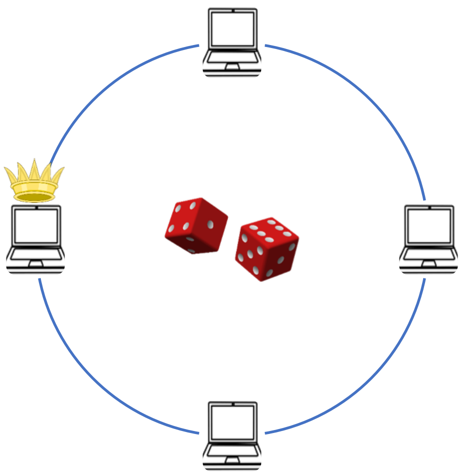
A game among a measure of *n* computer nodes:



Consensus game

A game among a measure of *n* computer nodes:

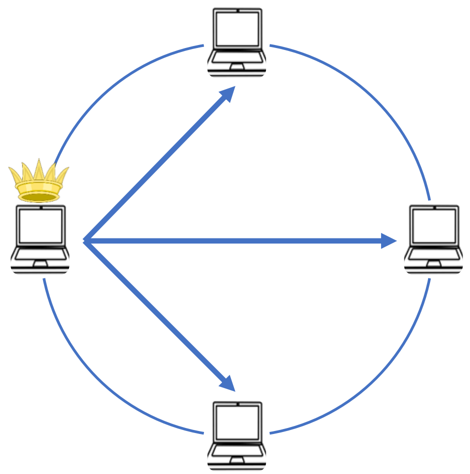
- Nature randomly selects one node as the ***leader***;
  - Denote all other nodes as ***backups***;



Consensus game

A game among a measure of *n* computer nodes:

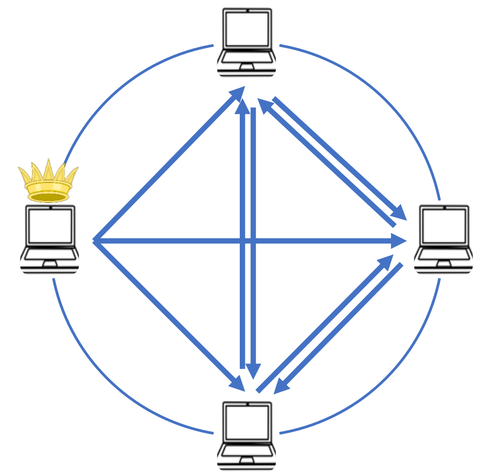
- Nature randomly selects one node as the ***leader***;
  - Denote all other nodes as ***backups***;
- Leader decides, for each backup, whether to ***send*** a message;
  - e.g. new batch of transactions in a blockchain;



Consensus game

A game among a measure of *n* computer nodes:

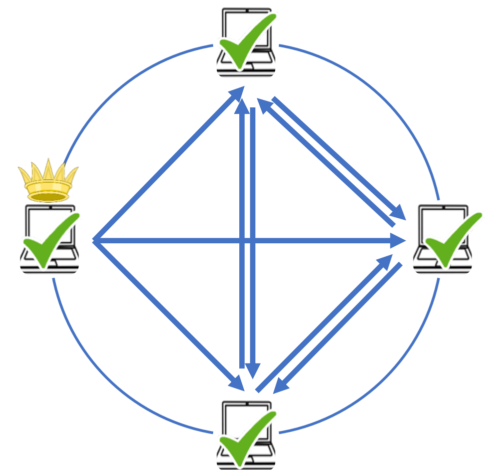
- Nature randomly selects one node as the ***leader***;
  - Denote all other nodes as ***backups***;
- Leader decides, for each backup, whether to ***send*** a message;
  - e.g. new batch of transactions in a blockchain;
- Each backup receiving message from leader
  - decides, for each other node, whether to ***forward*** message;



Consensus game

A game among a measure of *n* computer nodes:

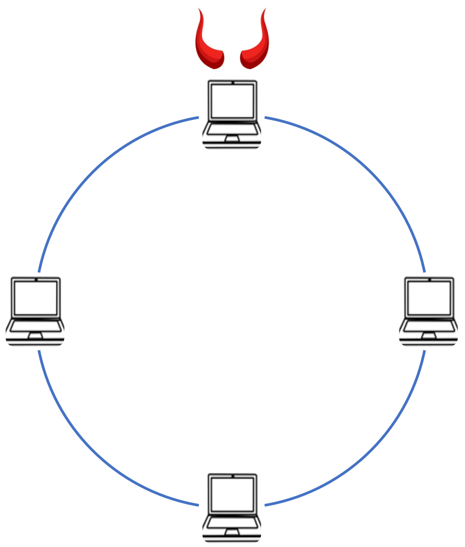
- Nature randomly selects one node as the ***leader***;
  - Denote all other nodes as ***backups***;
- Leader decides, for each backup, whether to ***send*** a message;
  - e.g. new batch of transactions in a blockchain;
- Each backup receiving message from leader
  - decides, for each other node, whether to ***forward*** message;
- Each node then decides whether to ***commit*** message
  - based on its local information set.



For simplicity, we study one round of synchronous peer communication in a single view. Lamport, Shostak and Pease (1982) study *f* rounds. Castro and Liskov (1999) (PBFT) study two rounds of communication with view changes. We also assume adequately close message delivery speeds to justify simultaneous moves in each step.

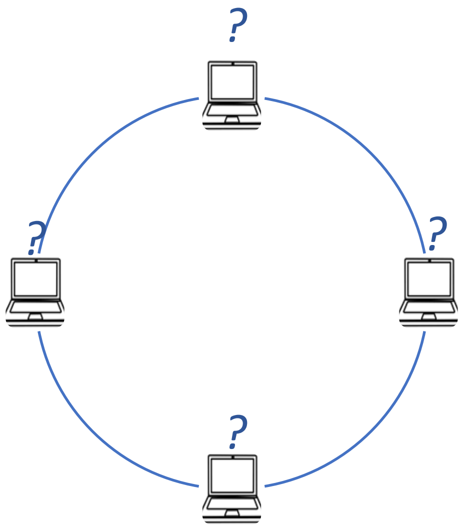
Two types of nodes

- Measure f of **Byzantine** nodes, who may take arbitrary actions;



Two types of nodes

- Measure f of **Byzantine** nodes, who may take arbitrary actions;



Two types of nodes

- Measure *f* of **Byzantine** nodes, who may take arbitrary actions;
- Measure *n−f* of **rational** nodes, who maximize utilities:

if consensus on message

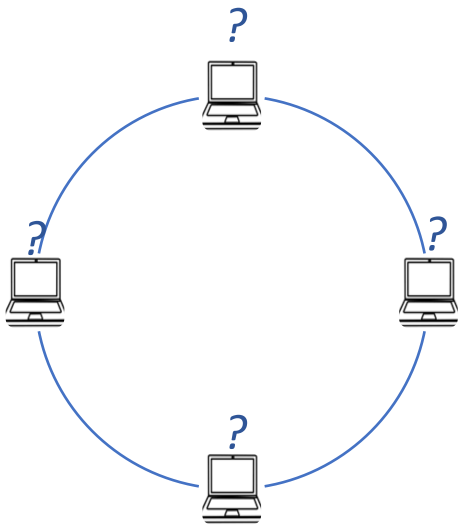
	Succeeds	Fails
Commit message	R > 0	-c < 0
Not commit message	0	0

Consensus succeeds iff "almost all" (measure *n−f*) rational nodes commit

- A dynamic game of imperfect info. ("coordination" & "cheap talk")

Solution concept:

- perfect Bayesian eqm + multi-priors over Byzantine strategies



Ambiguity aversion

- Rational nodes are ambiguity averse towards Byzantine strategies
  - "assume worst case scenario"

- Formally, a rational node i at any information set I_i facing all possible Byzantine strategies in $\mathcal{B}$ chooses action a_i to maximize

$$\min_{B \in \mathcal{B}} \mathrm{E}[u_i(a_i, A_{-i}; B) | I_i]$$

Characterizing all symmetric equilibria

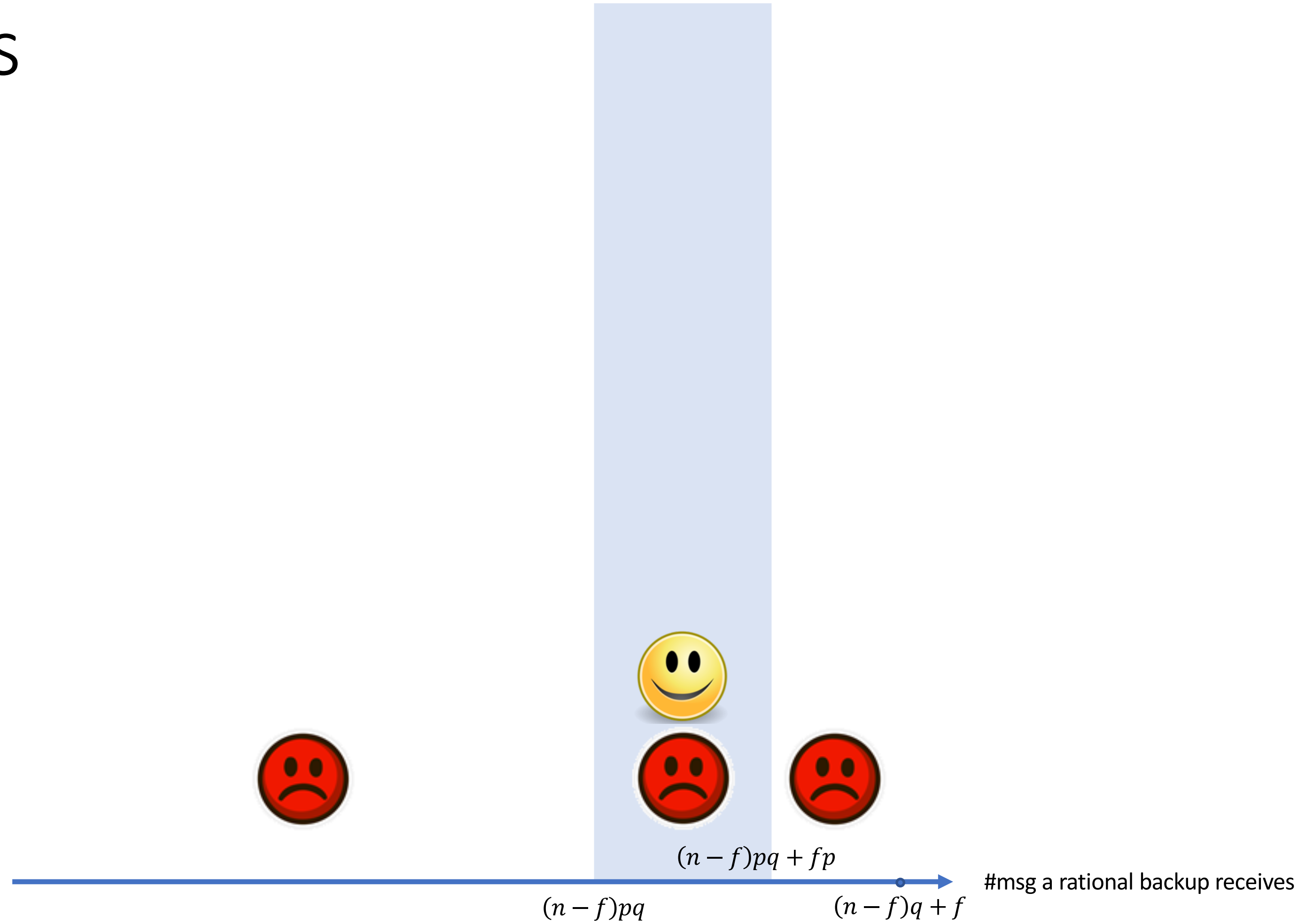
Consider a **candidate symmetric equilibrium** in which

- a rational leader sends message to each backup with prob. p
- a rational backup forwards message (if received) with prob. q
- a backup commits iff receiving
  $k \in E^1 \subset [0, (n - f)\, q + f]$ messages, with one from the leader; or
  $k \in E^0 \subset [0, (n - f)\, q + f]$ messages, none of which is from the leader.

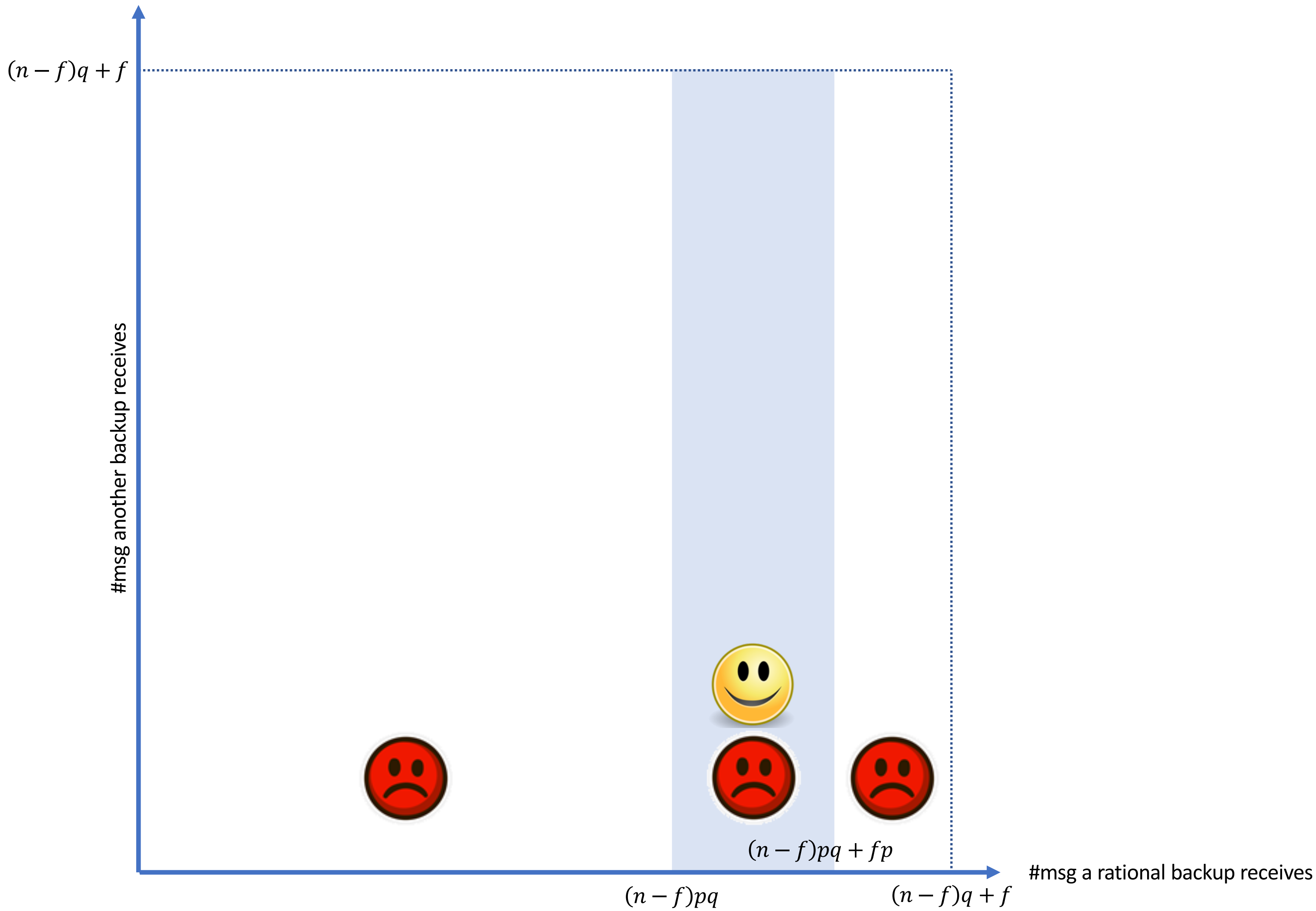
- If $E^1 \cup E^0 = \emptyset$, then a *gridlock equilibrium* (failed consensus)
- Our interest is in (successful) *consensus equilibrium* with $p > 0$ and $q > 0$

Inferences

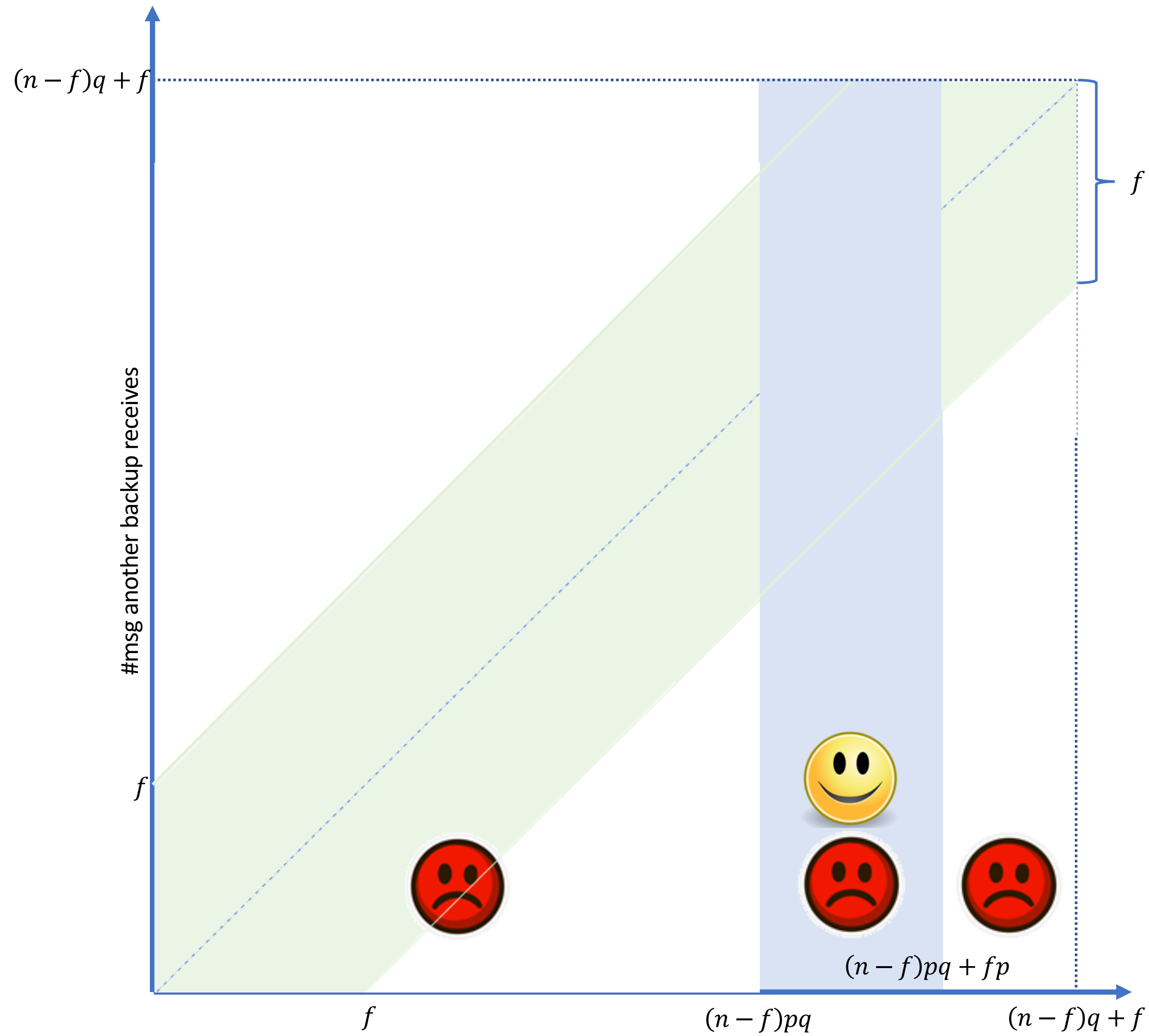
about the leader



about other rational nodes

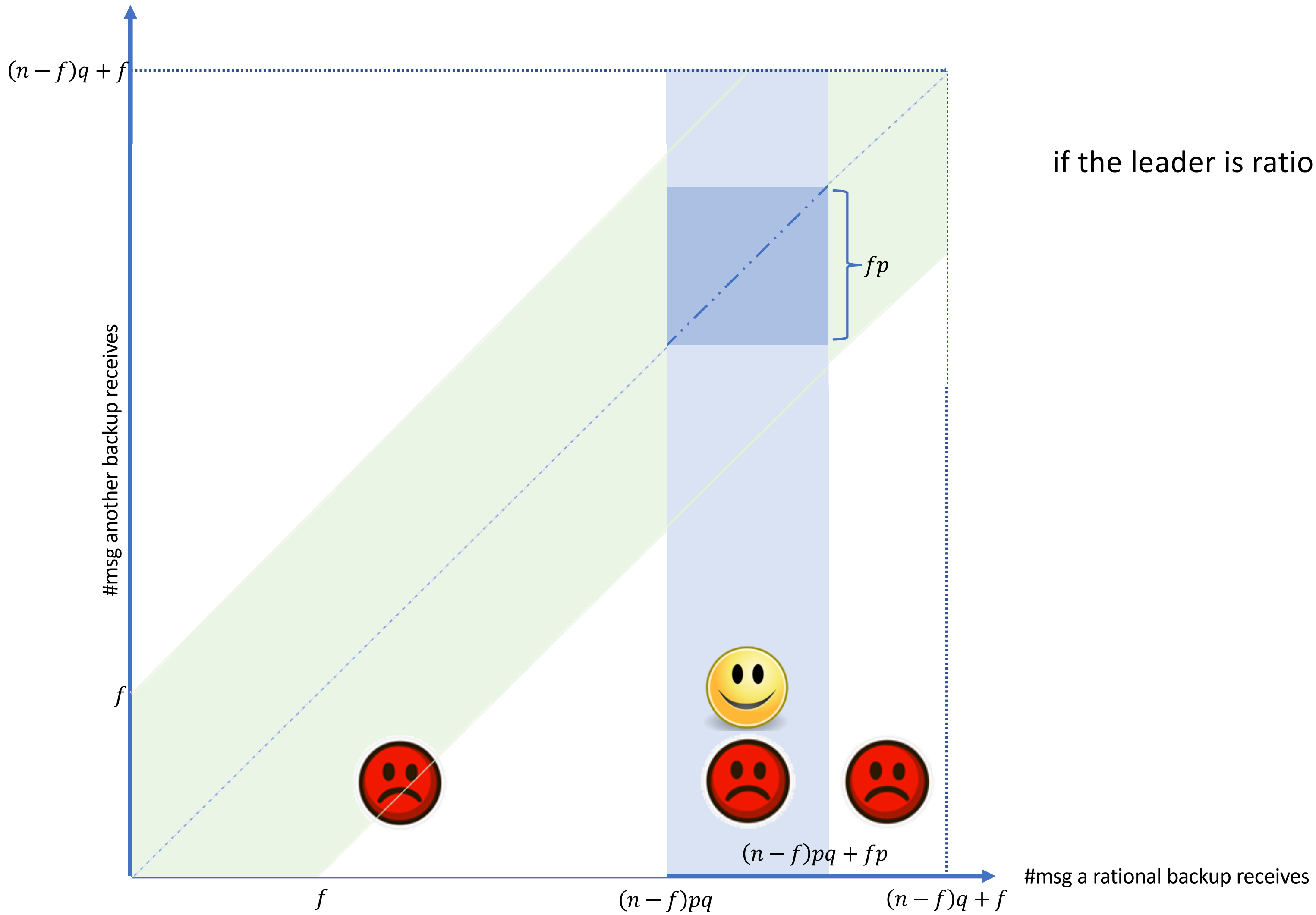


about other rational nodes



if the leader is Byzantine

- Byzantine nodes can coordinate:
- Deliver any number of msgs in this range with perfect precision



if the leader is rational

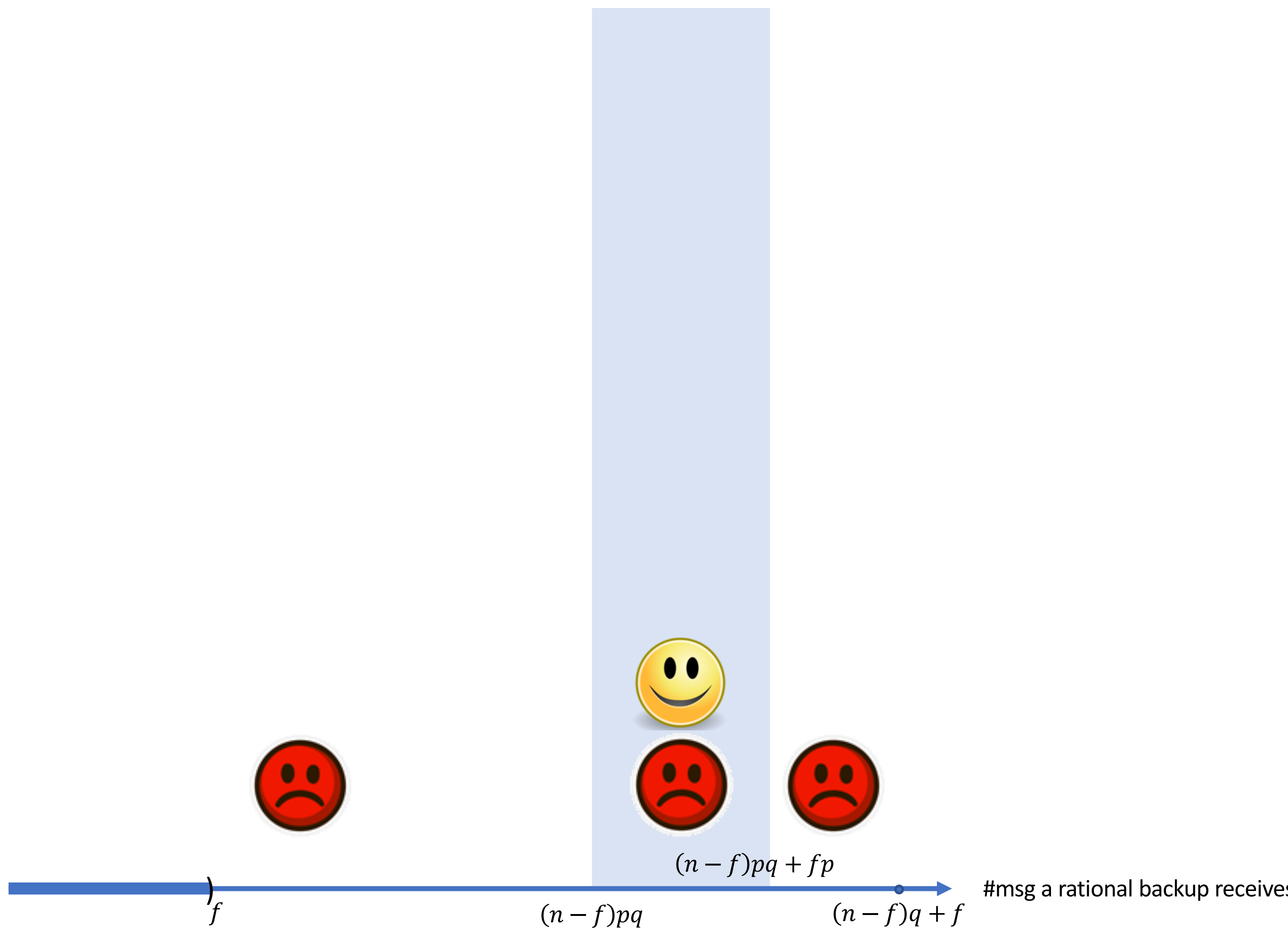
The properties of a consensus equilibrium

A consensus equilibrium has

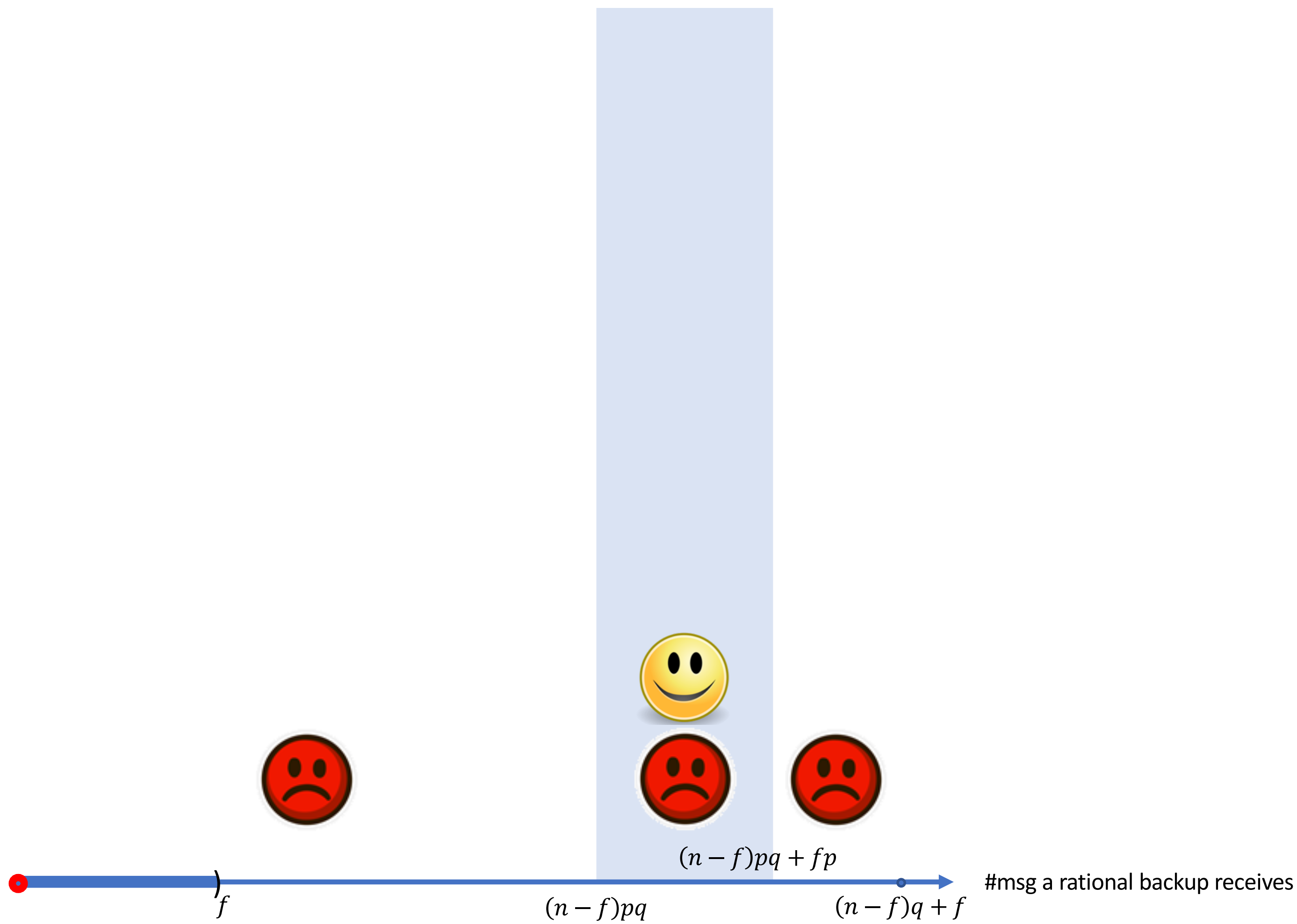
$$E^1 \cup E^0 = [(n-f)pq, (n-f)pq + fp]$$

A rational backup who knows the leader is Byzantine

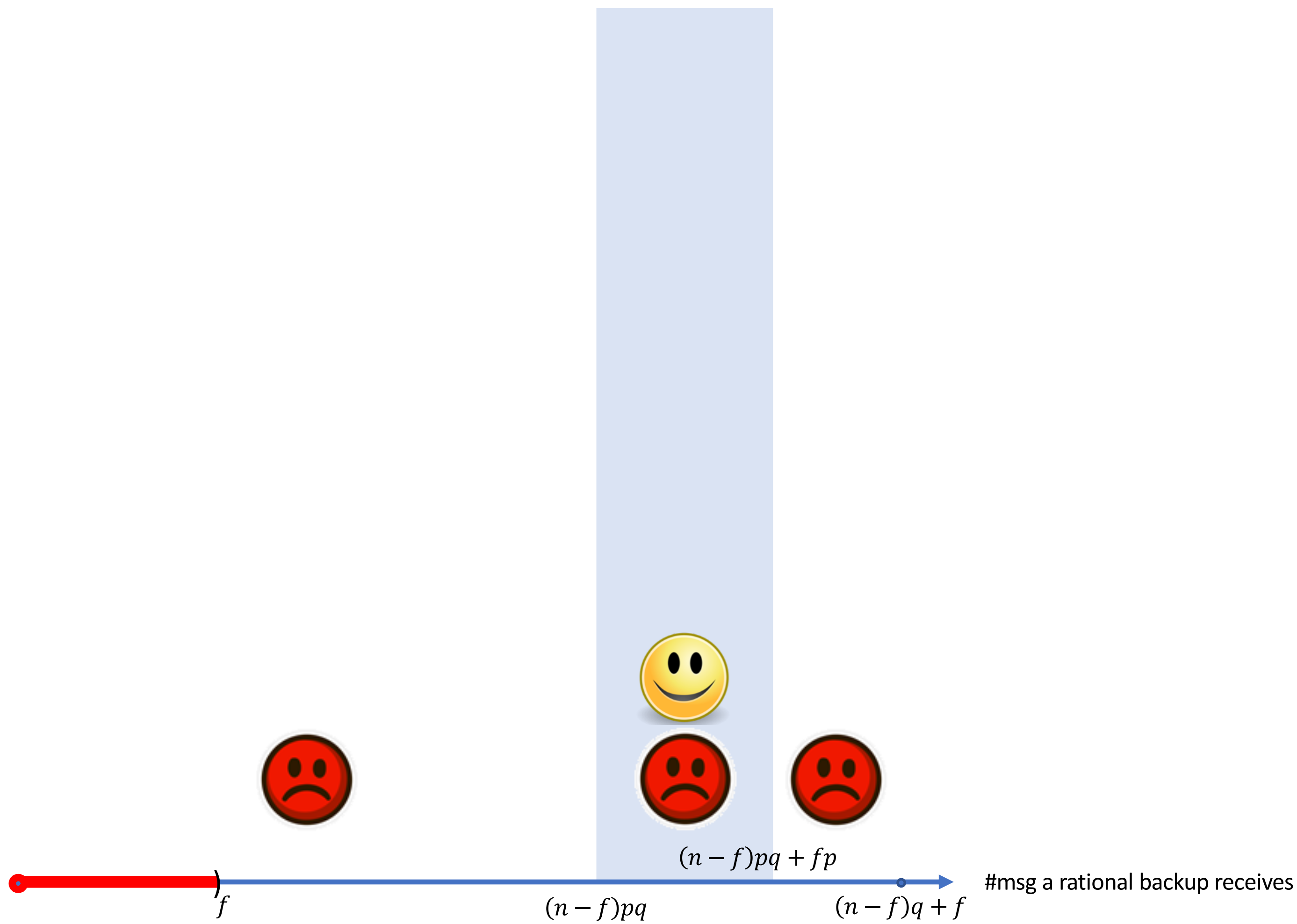
- always gets $-c$ from committing…
- thus does not commit
- except for when $p = 1$ and she receives exactly $k = (n-f)q + f$ messages



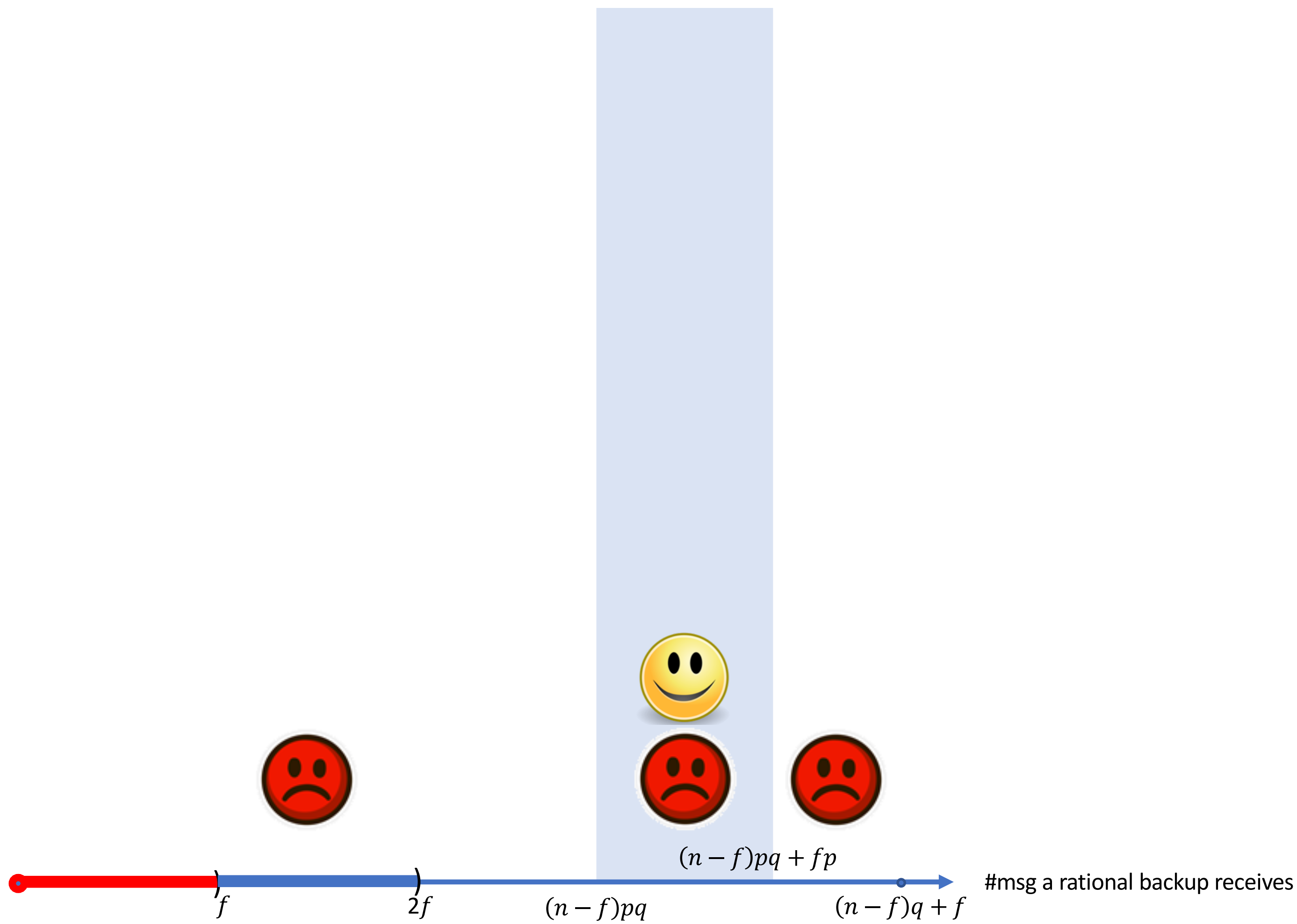
$(n-f)pq+fp$
f
$(n-f)pq$
$(n-f)q+f$
#msg a rational backup receives



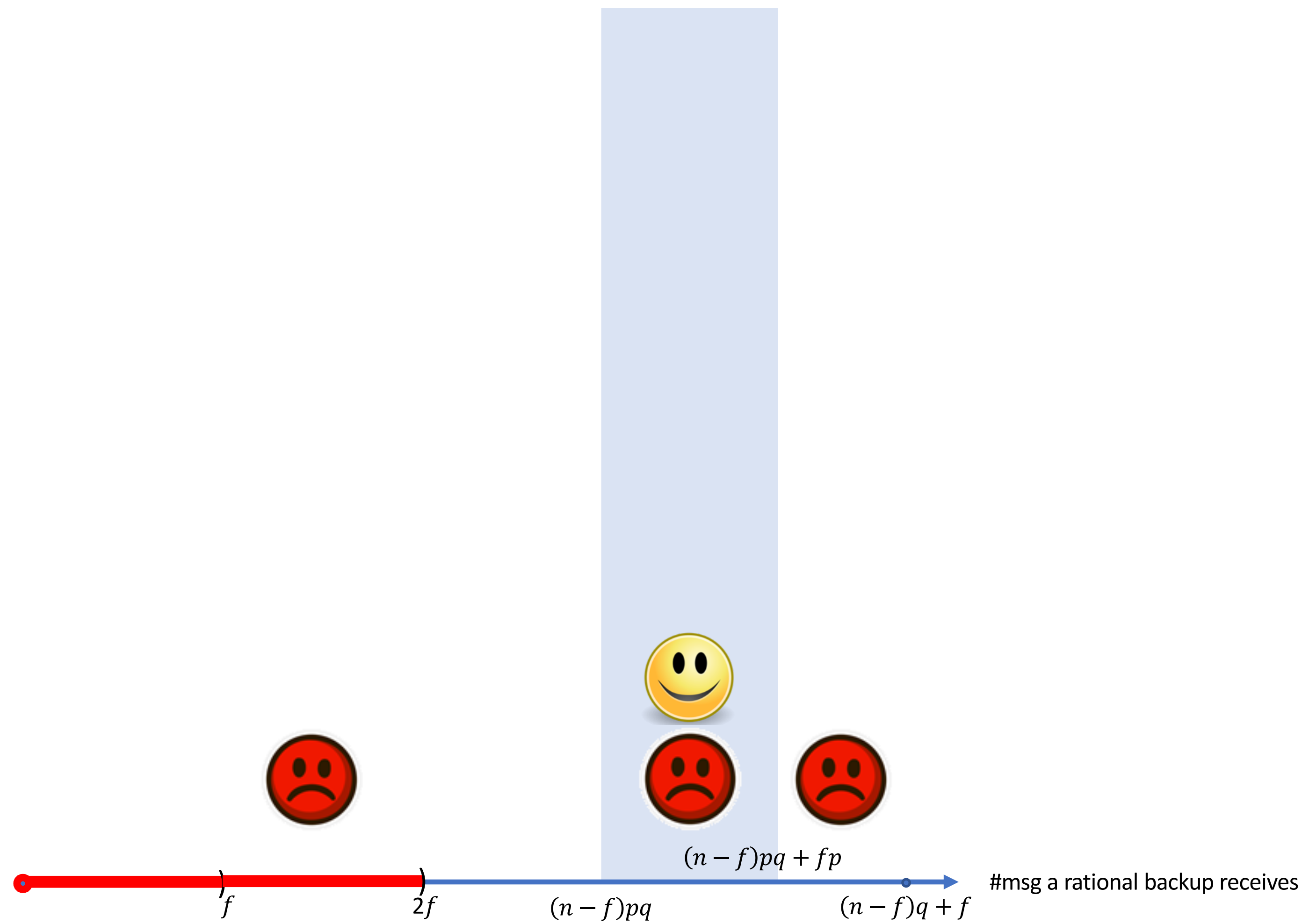
$(n-f)pq + fp$
f
$(n-f)pq$
$(n-f)q + f$
#msg a rational backup receives

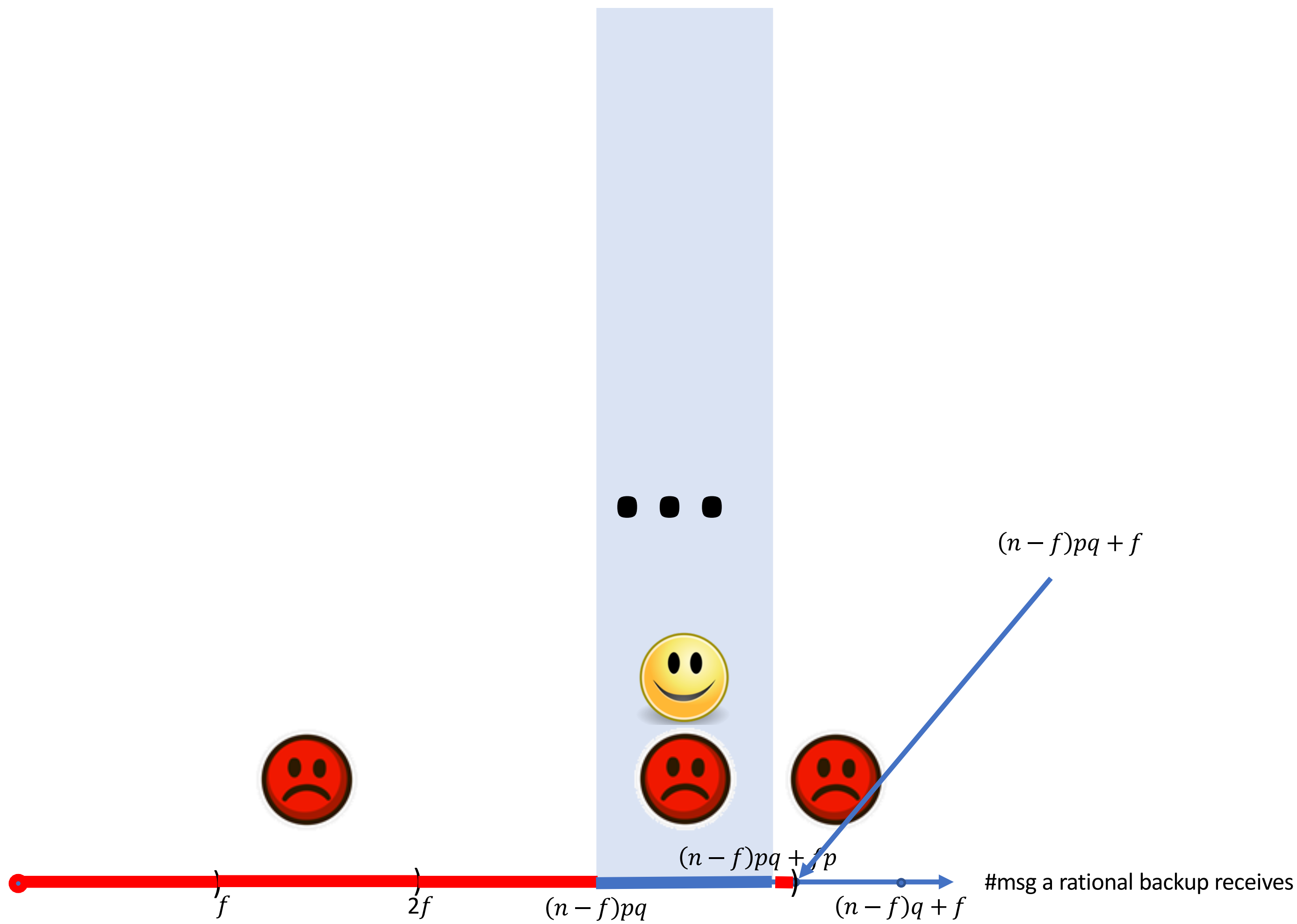


$(n-f)pq + fp$
f
$(n-f)pq$
$(n-f)q + f$
#msg a rational backup receives

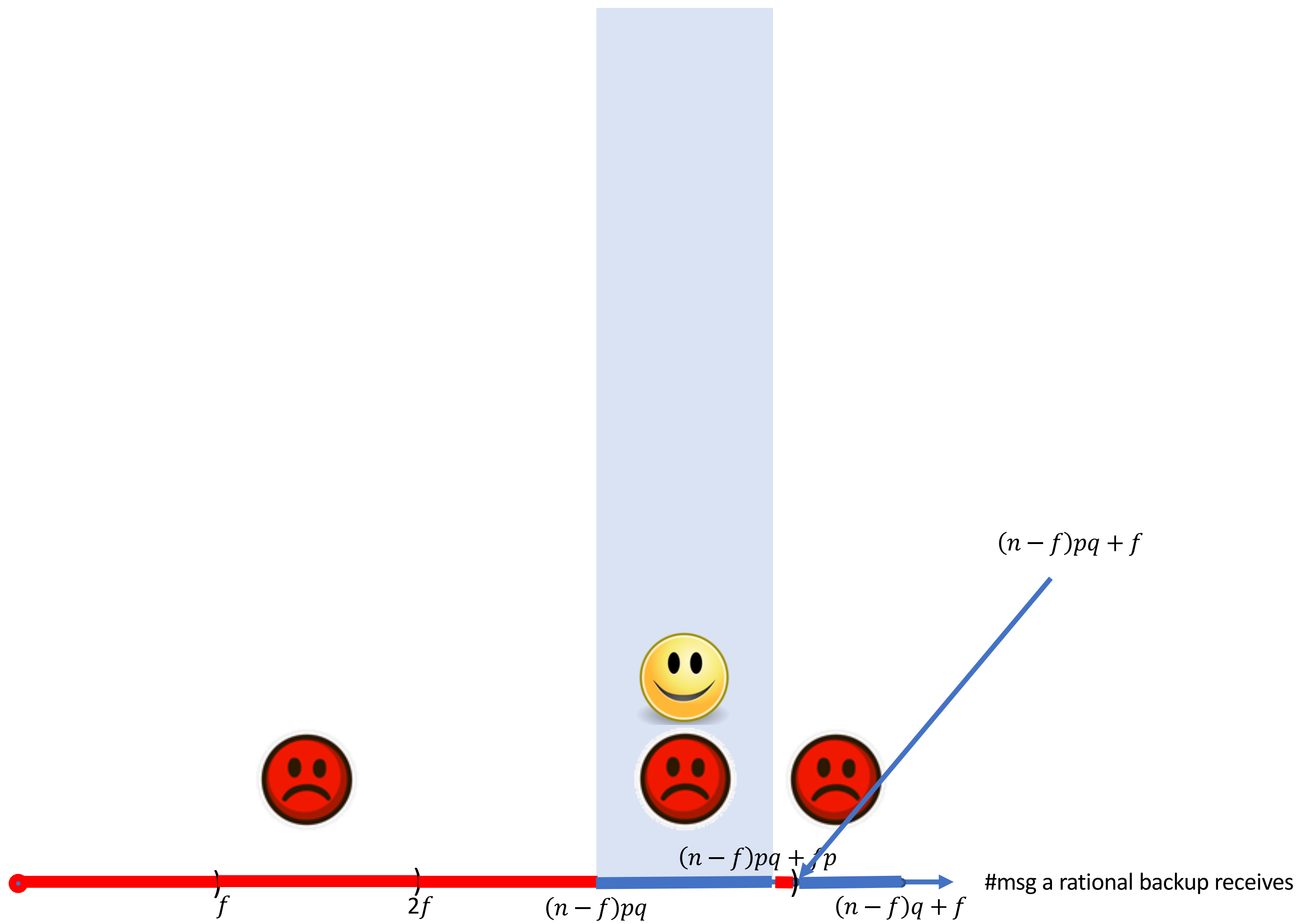


$(n-f)pq + fp$
#msg a rational backup receives
f
$2f$
$(n-f)pq$
$(n-f)q + f$

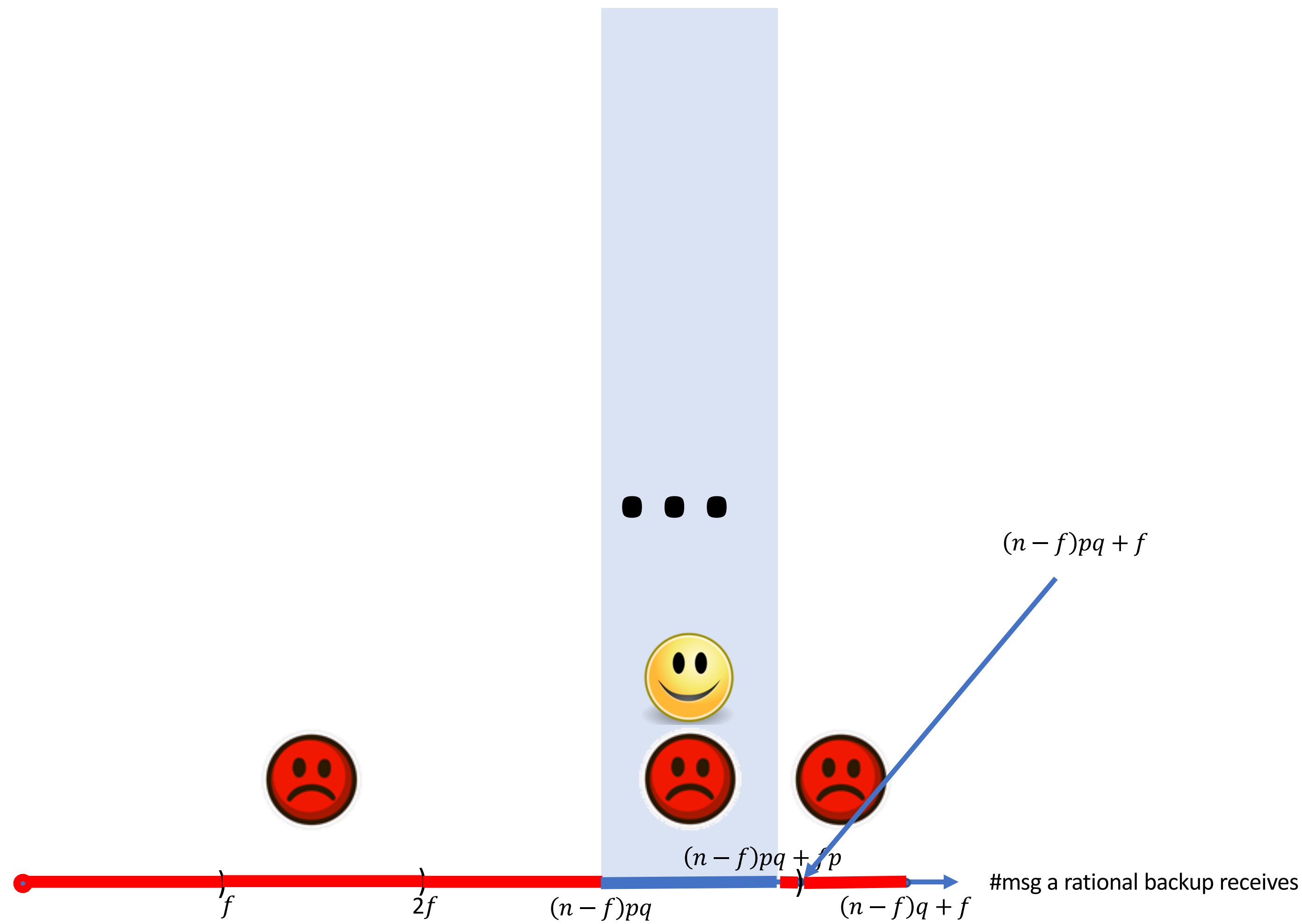

$(n-f)pq + fp$
#msg a rational backup receives
f
$2f$
$(n-f)pq$
$(n-f)q + f$



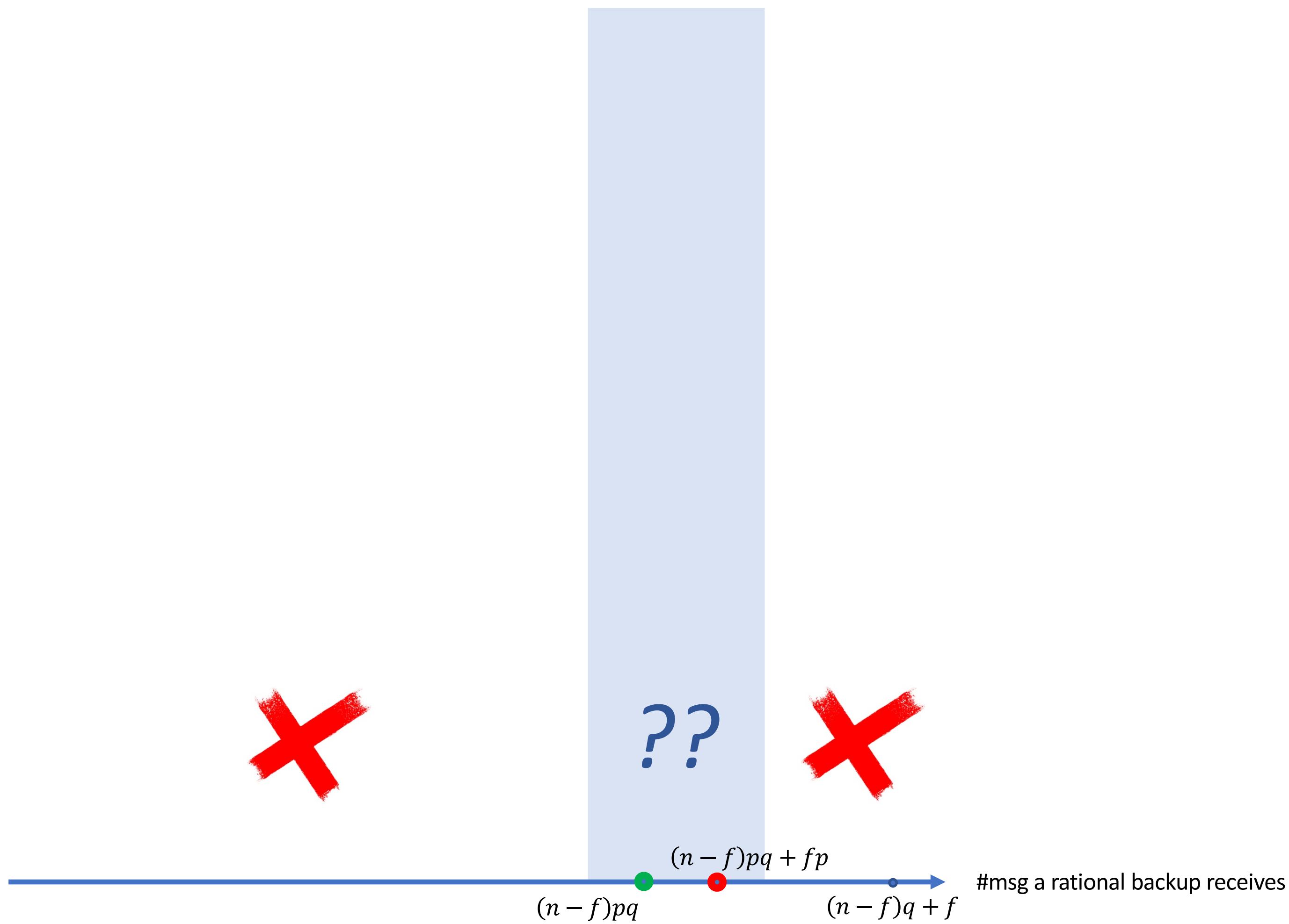
$(n-f)pq+f$
$(n-f)pq+fp$
#msg a rational backup receives
f
$2f$
$(n-f)pq$
$(n-f)q+f$

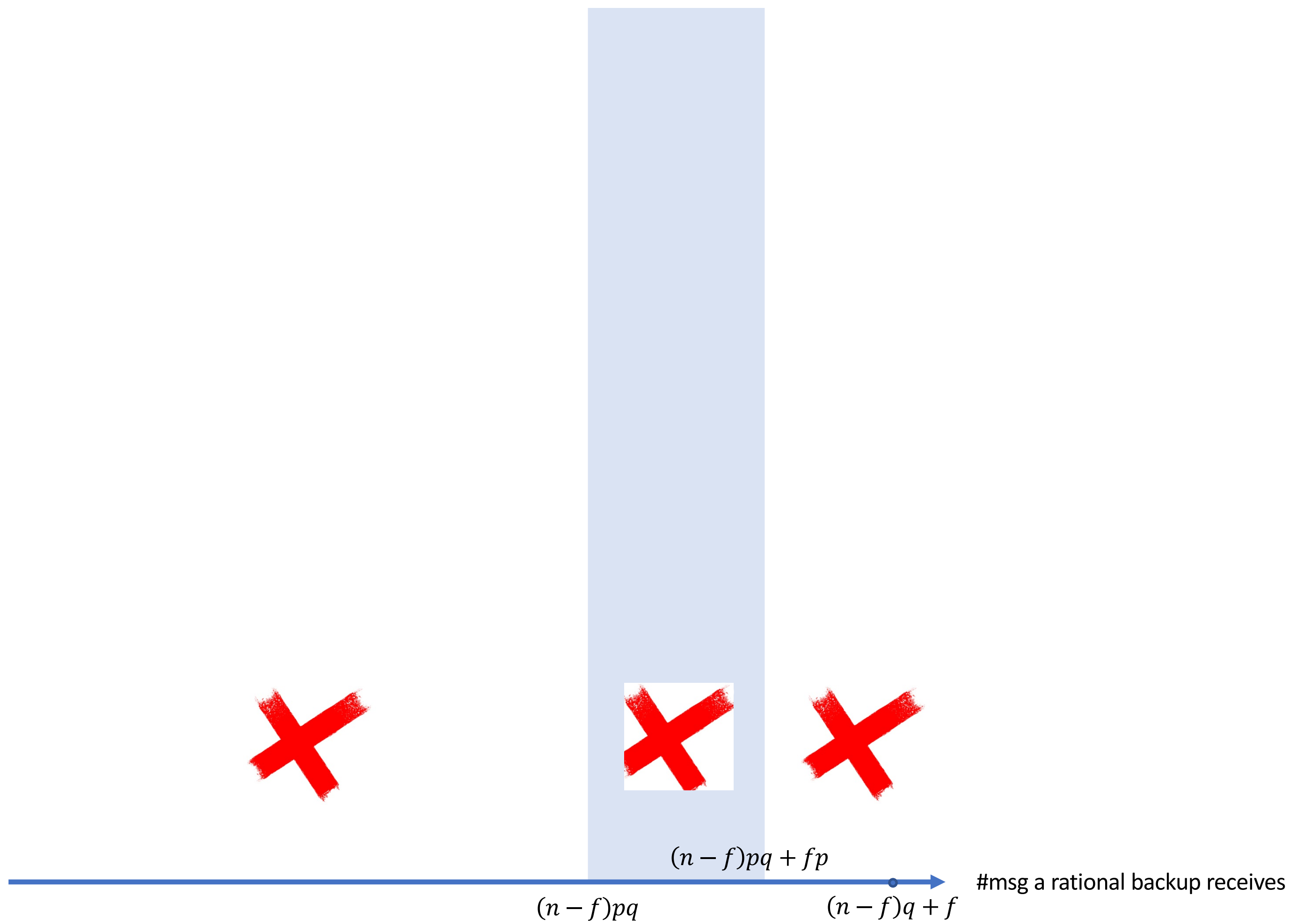


$(n-f)pq+f$
$(n-f)pq+fp$
#msg a rational backup receives
f
$2f$
$(n-f)pq$
$(n-f)q+f$



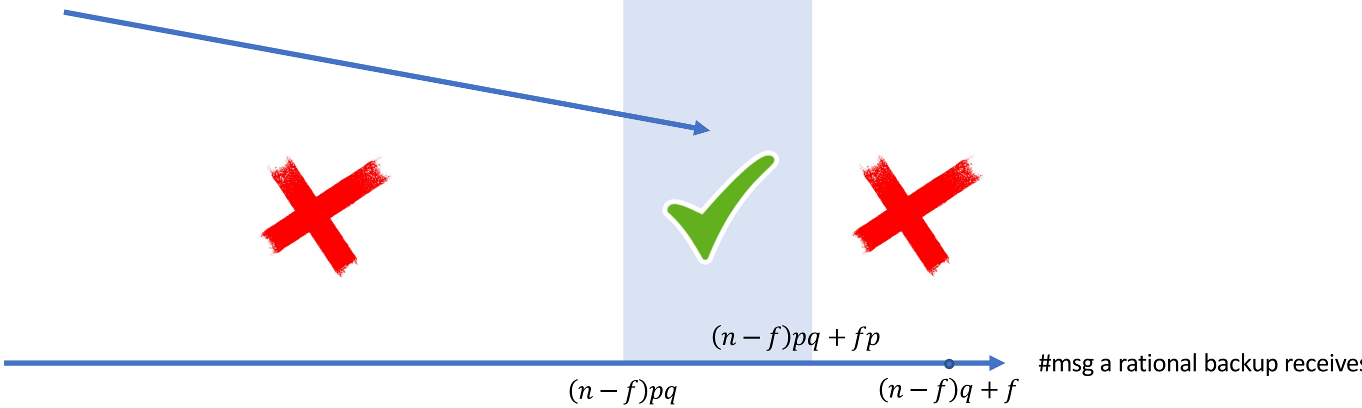
$(n-f)pq+f$
$(n-f)pq+fp$
#msg a rational backup receives
f
$2f$
$(n-f)pq$
$(n-f)q+f$


??
$(n-f)pq+fp$
#msg a rational backup receives
$(n-f)pq$
$(n-f)q+f$



$(n-f)pq + fp$
#msg a rational backup receives
$(n-f)pq$
$(n-f)q + f$

Need to characterize conditions for reward/penalty R and c under which it is indeed preferred to commit in this interval



All symmetric equilibria

- **A "gridlock" equilibrium**: discard communications and never commit
- **Singleton-$\boldsymbol{E^0}$-equilibria** indexed by $q \in (0,1]$ when $(n-f)R \geq fc$:
  - A rational leader sends message to each backup with $p = 1$;
  - A rational backup forwards message (if received) with prob. $q \in (0,1]$;
  - A rational backup commits *iff* receiving $k \in [(n-f)q, (n-f)q+f]$ messages, with one from the leader or $(n-f)q+f$ messages without any from the leader, i.e. $E^1 = [(n-f)q, (n-f)q+f]$ and $E^0 = \{(n-f)q+f\}$.
- **Interval-$\boldsymbol{E^0}$-equilibria** indexed by $p, q \in (0,1]$ when $\frac{1}{2}(n-f)R \geq fc$:
  - A rational leader sends message to each backup with prob. $p \in [\frac{fc}{(n-f)R}, 1-\frac{fc}{(n-f)R}]$;
  - A rational backup forwards message (if received) with prob. $q \in (0,1]$;
  - A rational backup commits *iff* receiving $k \in [(n-f)pq, (n-f)pq+fp]$ messages, receiving from the leader or not, i.e. $E^0 = E^1 = [(n-f)pq, (n-f)pq+fp]$

All symmetric equilibria

- **A "gridlock" equilibrium**: discard communications and never commit
- **Singleton-$\boldsymbol{E^0}$-equilibria** indexed by $q \in (0,1]$ when $(n-f)R \geq fc$:
  - A rational leader sends message to each backup with $p = 1$;
  - A rational backup forwards message (if received) with prob. $q \in (0,1]$;
  - A rational backup commits *iff* receiving $k \in [(n-f)q, (n-f)q + f]$ messages, with one from the leader or $(n-f)q + f$ messages without any from the leader, i.e. $E^1 = [(n-f)q, (n-f)q + f]$ and $E^0 = \{(n-f)q + f\}$.
- **Interval-$\boldsymbol{E^0}$-equilibria** indexed by $p, q \in (0,1]$ when $\frac{1}{2}(n-f)R \geq fc$:
  - A rational leader sends message to each backup with prob. $p \in [\frac{fc}{(n-f)R}, 1 - \frac{fc}{(n-f)R}]$;
  - A rational backup forwards message (if received) with prob. $q \in (0,1]$;
  - A rational backup commits *iff* receiving $k \in [(n-f)pq, (n-f)pq + fp]$ messages, receiving from the leader or not, i.e. $E^0 = E^1 = [(n-f)pq, (n-f)pq + fp]$

Equilibria and Blockchain Protocol

- What does an existence (or not) of an equilibrium mean for blockchain protocol design?
- If the protocol prescribes *p, q,* E^0, E^1, *c, R* s.t. *p, q,* E^0, E^1 is an equilibrium given *n, f, c* and *R*, then rational nodes have no incentive to deviate, and consensus is reached
- We can calculate the cost of incentives needed (*R, c*) to achieve consensus given *p*
  - *singleton-*E^0-eq'a require lower *R* than fractional-*p*-eq'a (for the same *c*)
  - for *interval-*E^0-eq'a, *p* further from ½ requires higher *R*

Interval-E^0 equilibria

- If message from the leader received, the expected payoff from committing:

$$\frac{p(n-f)}{p(n-f)+f}R+\frac{f}{p(n-f)+f}(-c)$$

- If message from the leader not received, the expected payoff from committing:

$$\frac{(1-p)(n-f)}{(1-p)(n-f)+f}R+\frac{f}{(1-p)(n-f)+f}(-c)$$

- For both to be positive, *p* cannot be too large or too small
  - $p \in [\frac{fc}{(n-f)R}, 1-\frac{fc}{(n-f)R}]$;
  - and only when $\frac{1}{2}(n-f)R \geq fc$

Message losses

- All messages sent are delivered with prob. α < 1
- A “gridlock” equilibrium still exists
- Singleton-E^0-equilibria no longer exist
- Interval-E^0-equilibria require higher R/c to sustain for small α
- Supporting R/c regions expands as message loss prob. α decreases

Why does it matter?

- Operational success of any blockchain depends on its design.
- Accounting for incentives in BFT consensus:
  - All designs are subject to multiple equilibria concerns
    - gridlock equilibria always exist ⇒ possibility of system stuck
  - Small probability of message loss significantly affects equilibria
  - Provides guidance on **cost of incentives** needed to achieve consensus
    - Less costly when protocol asks for sending message with p=1/2
    - Recommendation different from traditional BFT