Problem of agreement / Consensus

with Byzantine nodes

TD

Assumptions for Ex. 1 and 2

- Full network (reliable links) of 4 nodes, synchronous rounds
- At most 1 Byzantine node
- Specification:
  - Agreement: every two **non-faulty** processes decide on the same value.
  - *Validity*: if all **non-faulty** processes have the same initial value v, v is the only possible decision value of **non-faulty processes**.
  - *Termination:* all non-faulty processes eventually **decide** on a value.

Algorithm Ex. 1

states_i and start_i:

v_i: U; initialised to the initial value of i

m_i: U

msgs_i: *send v_i to everyone (including me)*

trans_i:

If *n messages with values (including mine) are received,* ***then***

Calculate the majority value m_i of these values (take the smallest one in case of a tie)

v_i:= m_i

EndIf

Algorithm Ex. 1.1

states_i and start_i:

v_i: U; initialised to the initial value of i

m_i: U

msgs_i: *send v_i to everyone (including me)*

trans_i:

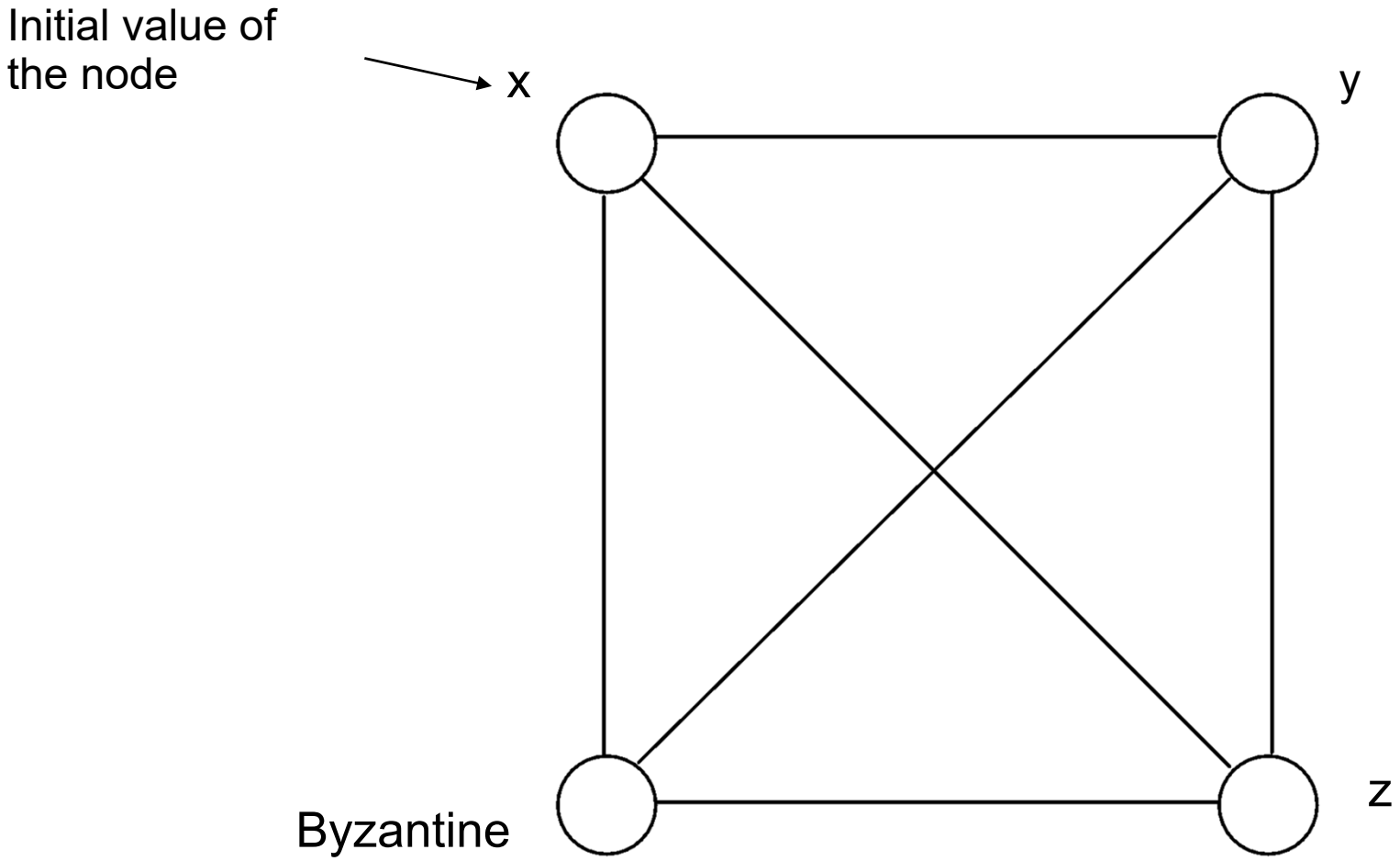
~~***If** n messages with values (including mine) are received, **then***~~

*Calculate the majority value m_i of the values **received in this round** (take the smallest in case of a tie)*

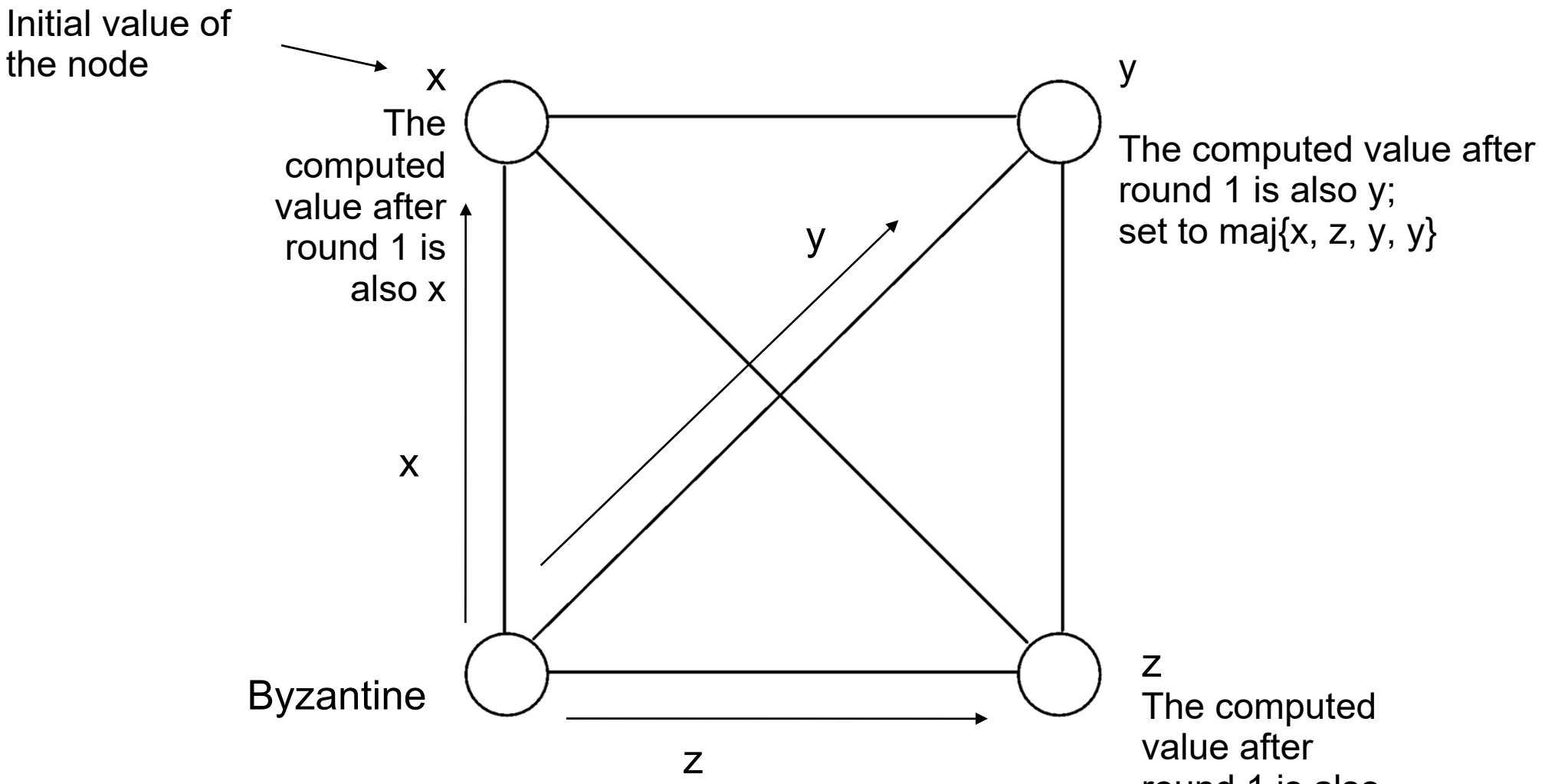
v_i:= m_i

~~***EndIf***~~

Counter-example Ex. 1.2

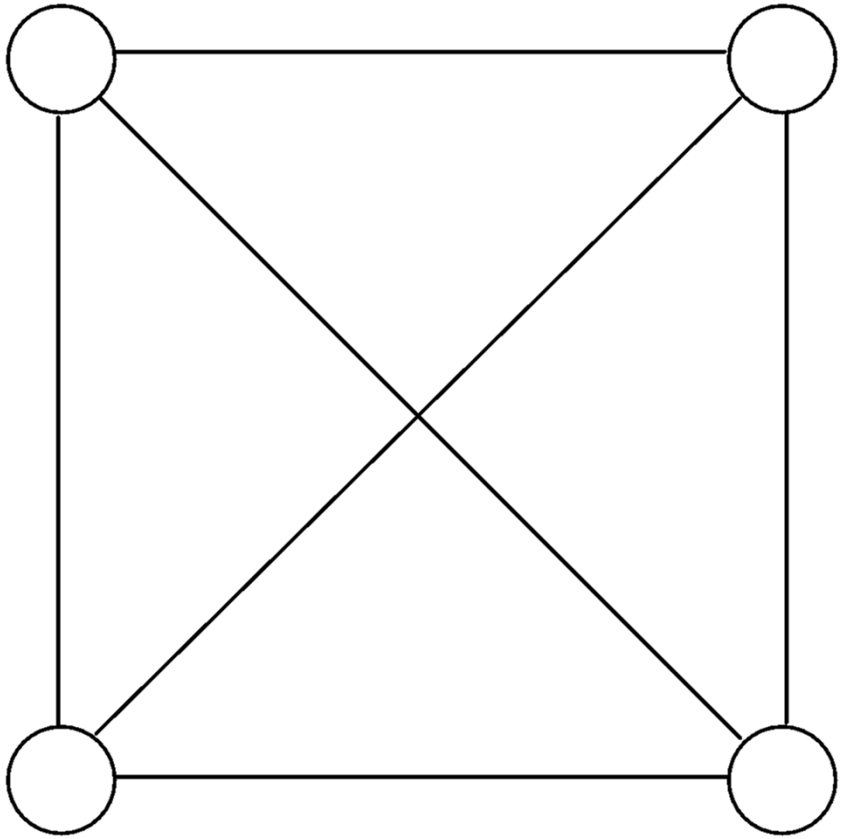


Counter-example e.g. 1.2



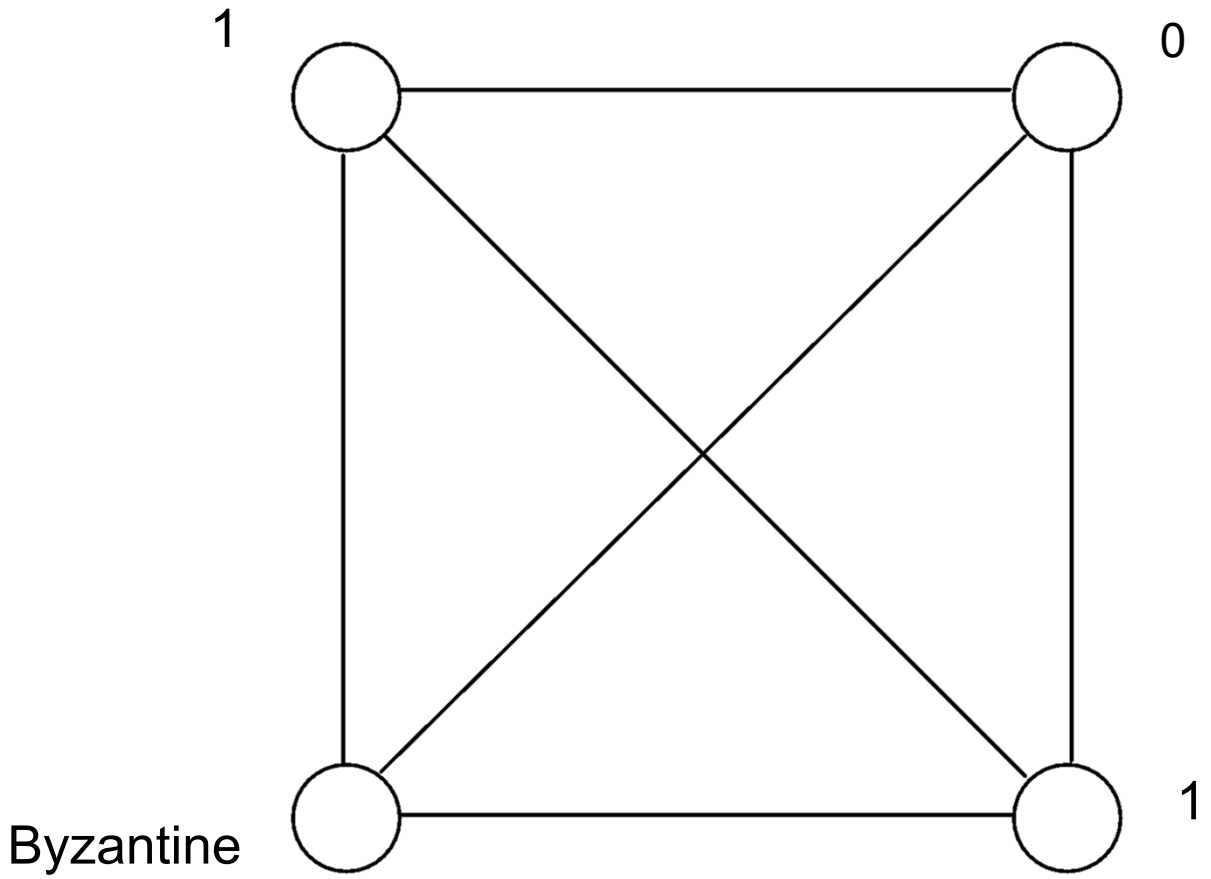
Messages sent in round 1 by a Byzantine node, and in each subsequent round

What about a binary consensus? Ex. 1.2



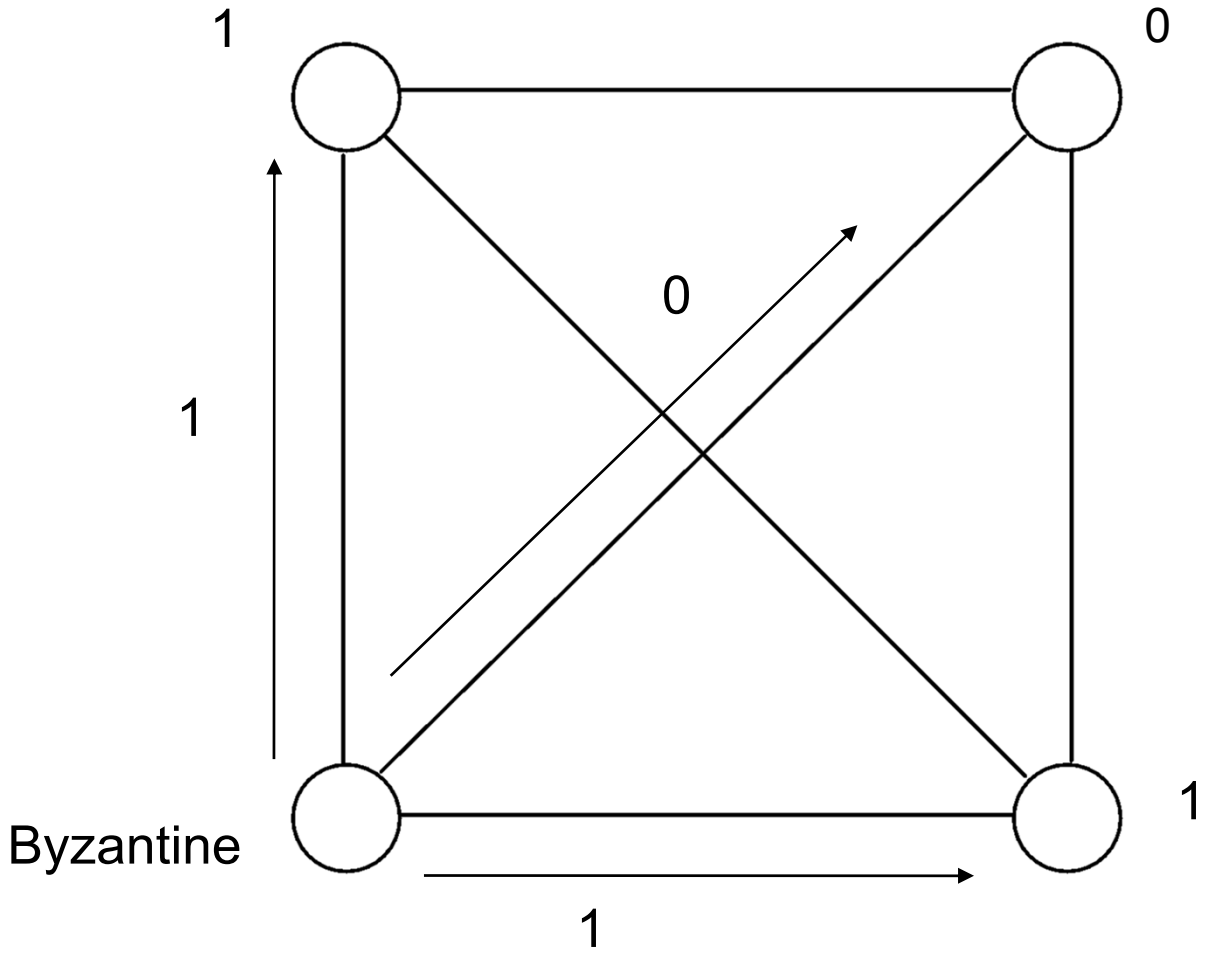
Byzantine

What about a binary consensus? Ex. 1.2



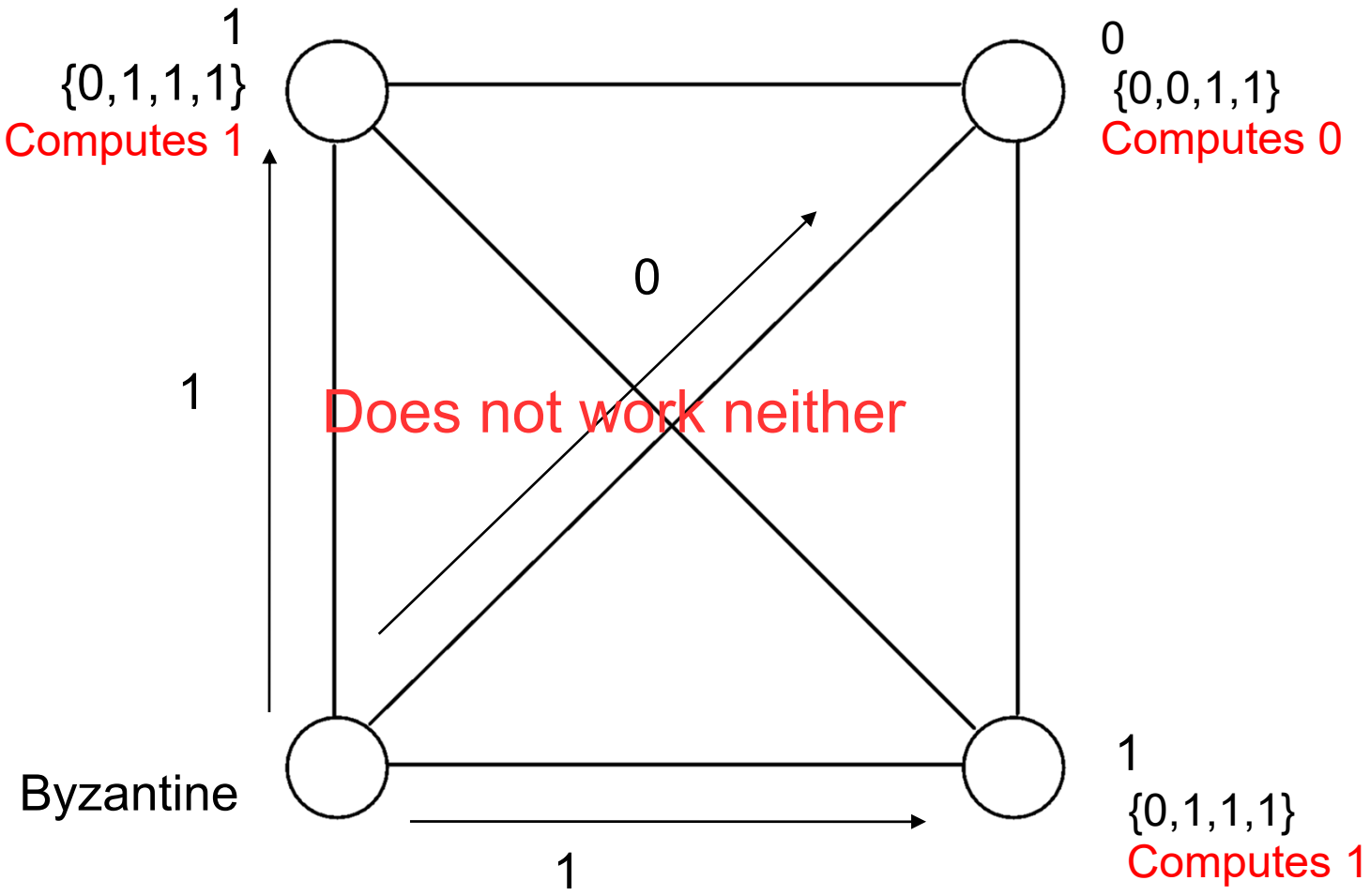
What about a binary consensus? Ex. 1.2

In round 1



What about a binary consensus? Ex. 1.2

In round 1



Repeat for each subsequent round

Algorithm Ex. 2

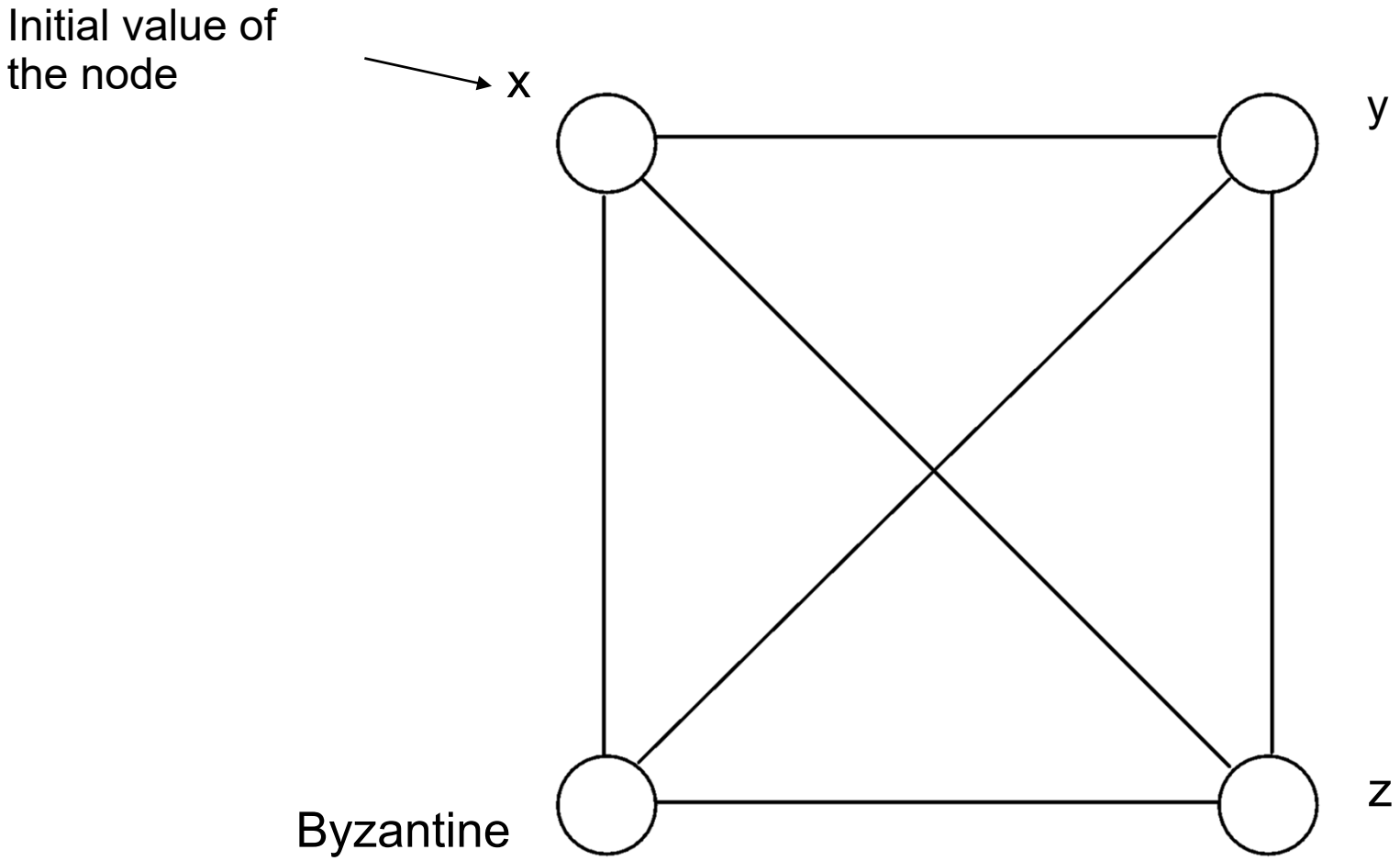
Round 1: *each process broadcasts its initial value to all the others and receives the values of the others. It calculates the average value of the values available to it, and chooses the value closest to the calculated average (if there are two close values, the higher of the two is chosen).*

Rounds 2 and 3: *Similar to the first, but sending the value chosen in the previous round.*

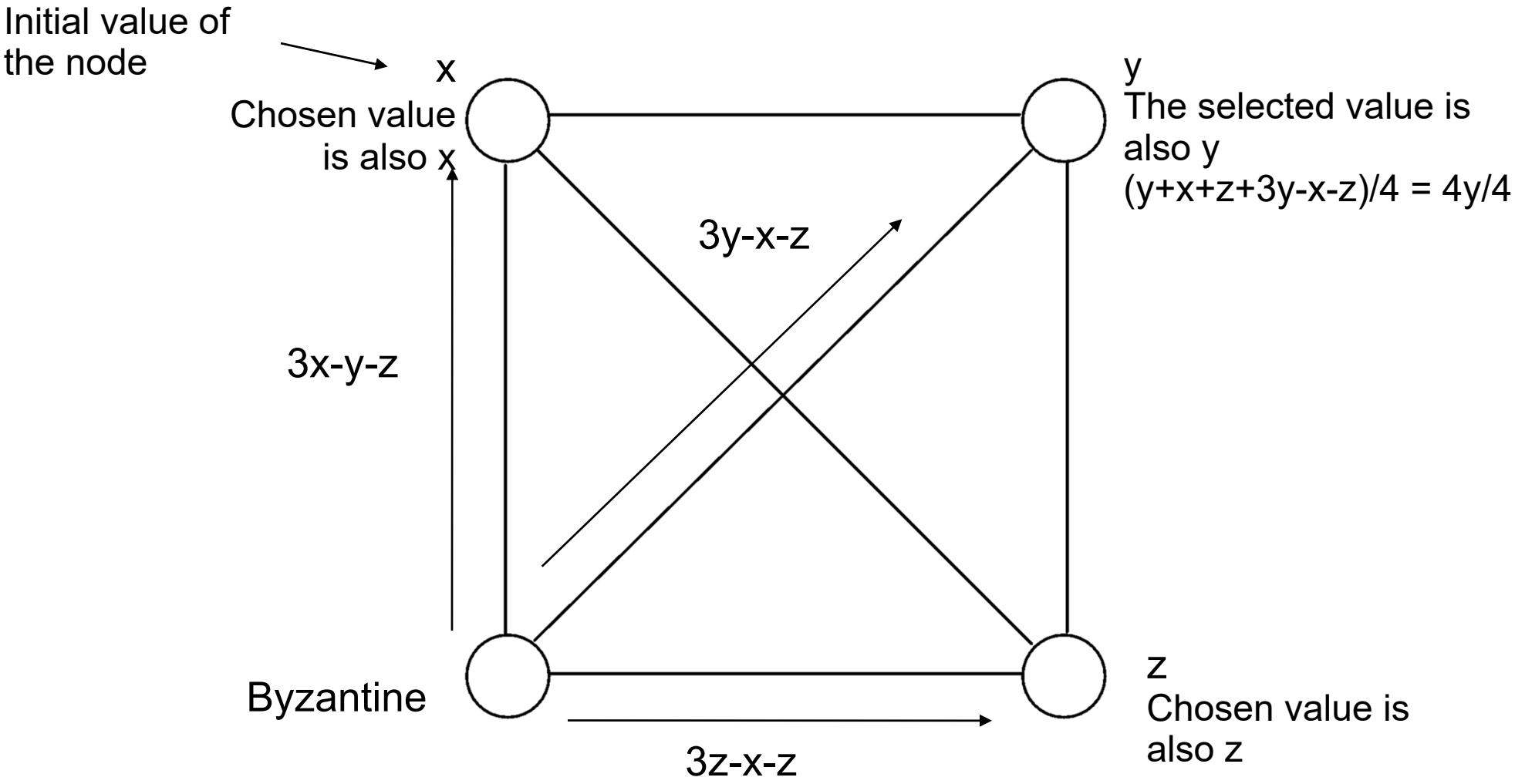
The decision value is the value chosen at the end of the third round.

Is this protocol a correct solution to the consensus problem?

Counter-example Ex. 2

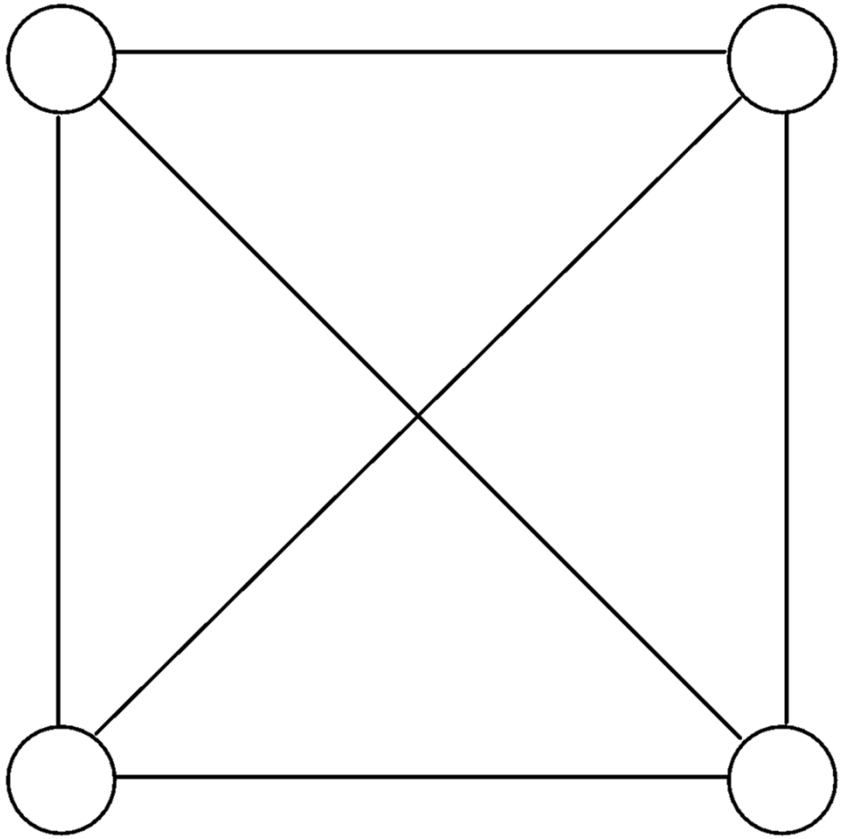


Counter-example Ex. 2



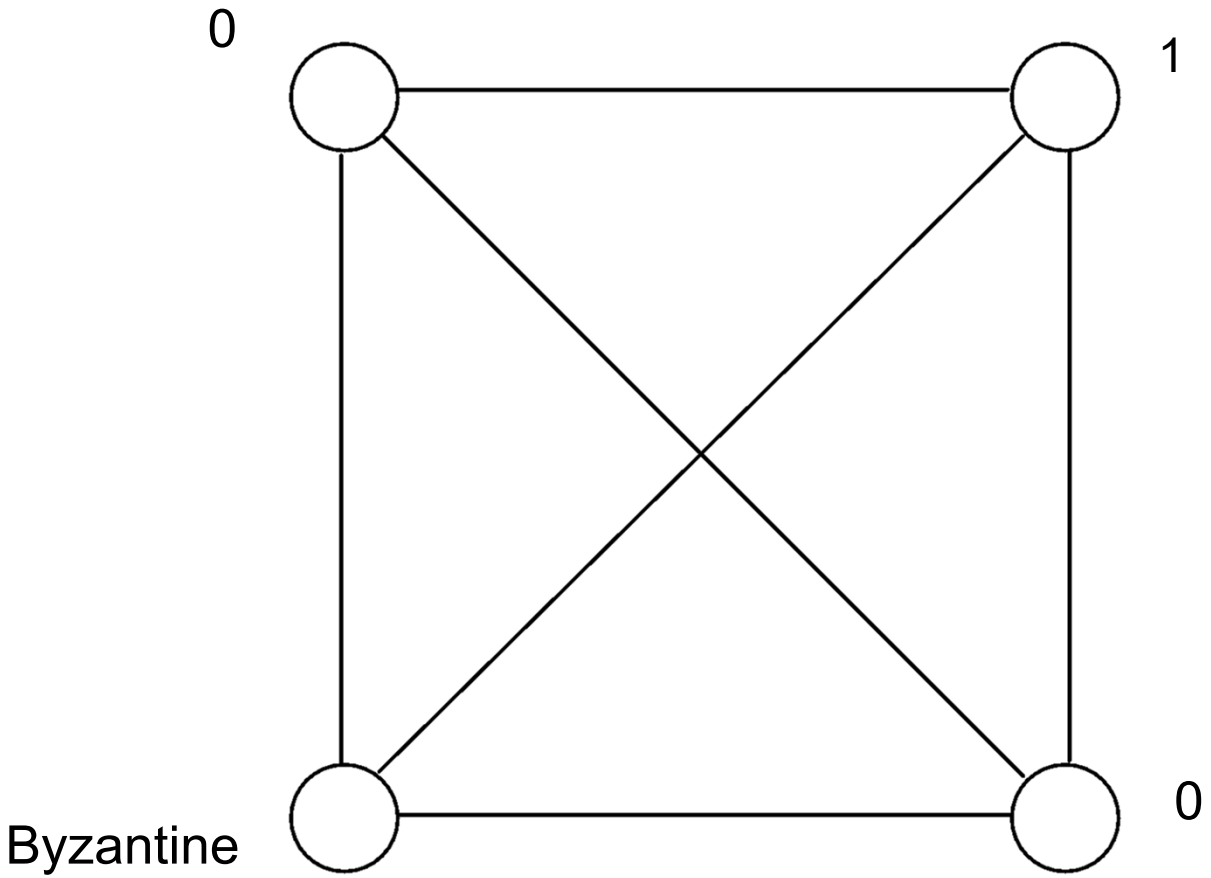
Messages sent in round 1 by a Byzantine node, and in each subsequent round

What about the binary consensus? Ex. 2

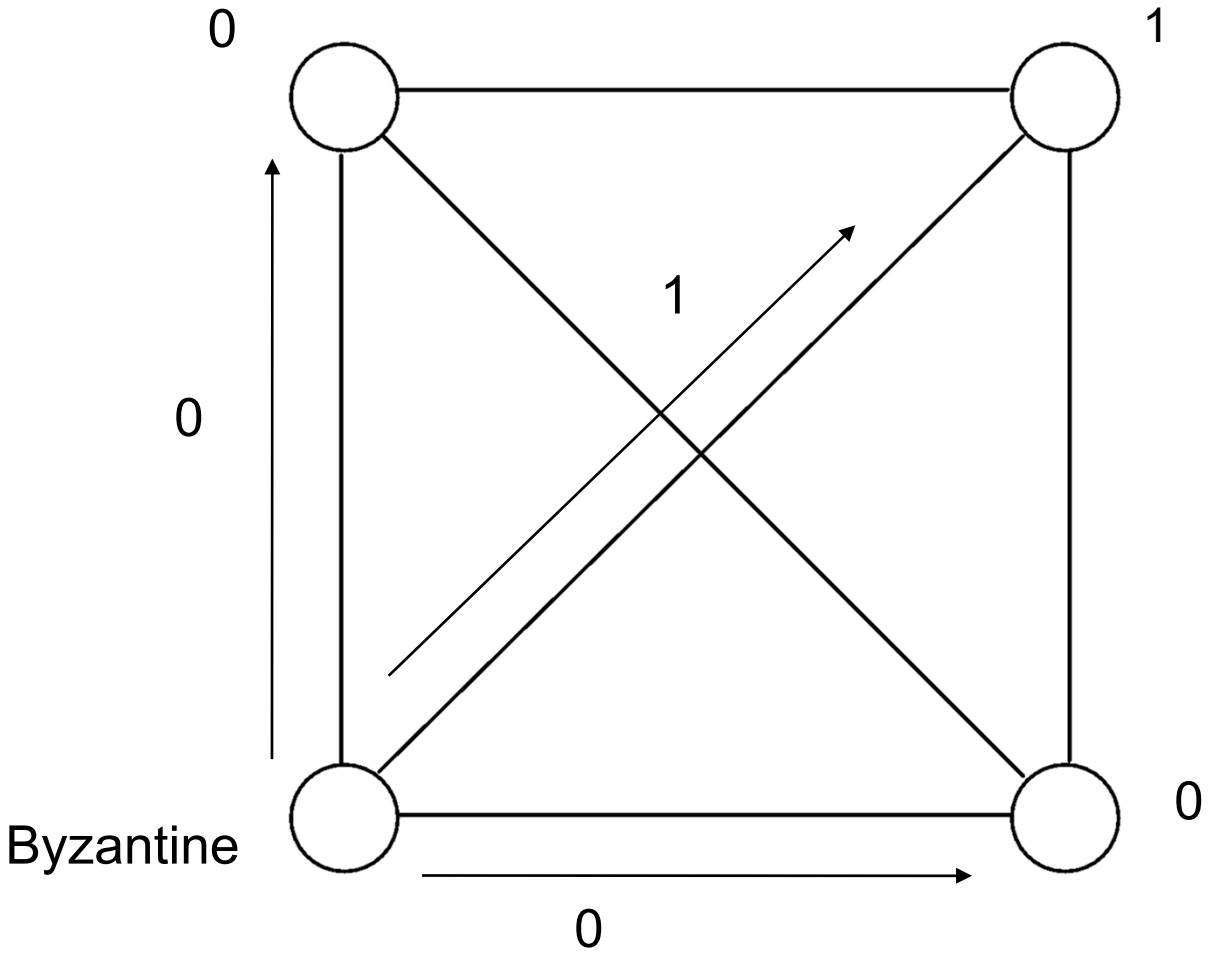


Byzantine

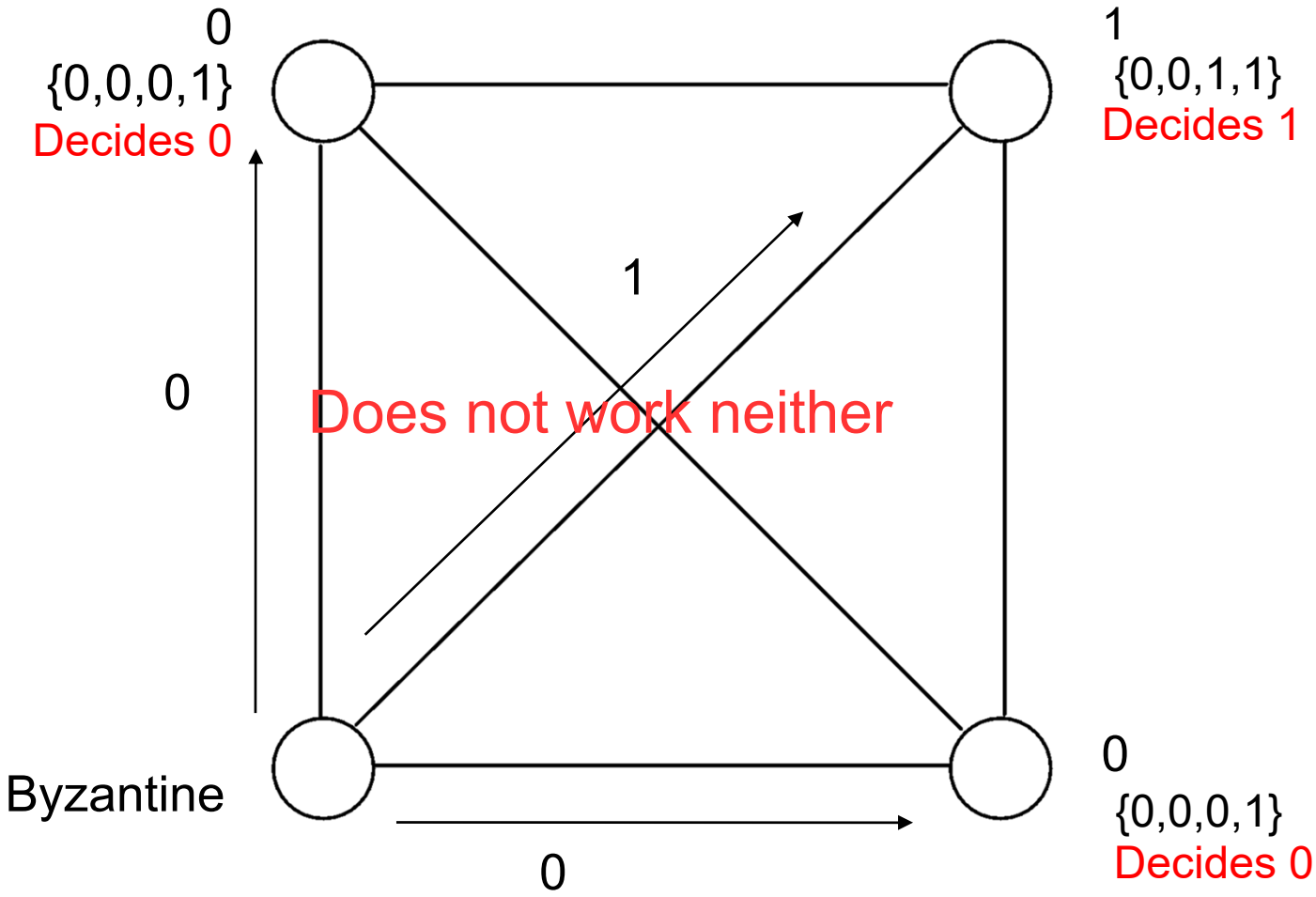
What about the binary consensus? Ex. 2



What about the binary consensus? Ex. 2



What about the binary consensus? Ex. 2



The problem of reliable broadcast with Marshal

with a Byzantine node

TD - Ex. 3

Assumptions Ex. 3

- Full network (reliable links) of 4 nodes, synchronous rounds
- At most one 1 Byzantine
- One of the 4 nodes is called a **marshal** (a distinguished process)
- **Purpose:** to broadcast the *initial value of the marshal* to all (correct) nodes and agree on this value.

Notes:

- The marshal can be Byzantine
- “Distinguished” means that the other nodes know which node is the marshal

Reliable **broadcast** – specification

- *Termination*: Every non-faulty general eventually decides/accepts a value.
- *Validity*: If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.
- *Agreement*: Any two non-faulty generals never decide/accept different values.

Algorithm

First round: *the marshal broadcasts his value to all;*

Second round: *each node other than the marshal sends the value received in round 1 to all the others except the marshal*

- **At the end of the second round**: *each process decides (accepts) the median value among the values available to it.*
  (A received empty message – null – is considered to be a default value u' ∈ U.)

Note: if the Byzantine is not the marshal and sends messages in the first round, they are ignored by the other nodes (the marshal being distinguished).

Proof of correctness Ex. 3

The 3 conditions of the specification must be proven:

1) **Termination** (*Any non-faulty general eventually decides/accepts a value.*)
2) **Validity** (*If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.*)
3) **Agreement** (*Any two non-faulty generals never decide/accept different values.)*

Proof of correctness: Termination

Termination: *Any non-faulty general eventually decides/accepts a value.*

Proof of correctness: Termination

Termination: *Any non-faulty general eventually decides/accepts a value.*

→ Trivial: the algorithm finishes after two rounds and each correct node does decide on a value at the end of these rounds (see algorithm)

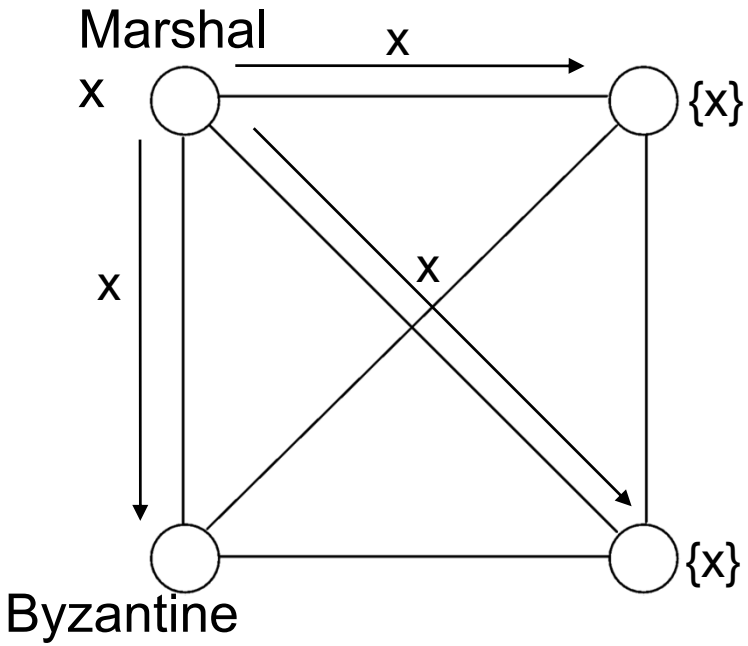
Proof of correctness: Validity

Validity: *If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.*

Proof of correctness: Validity

Validity: *If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.*

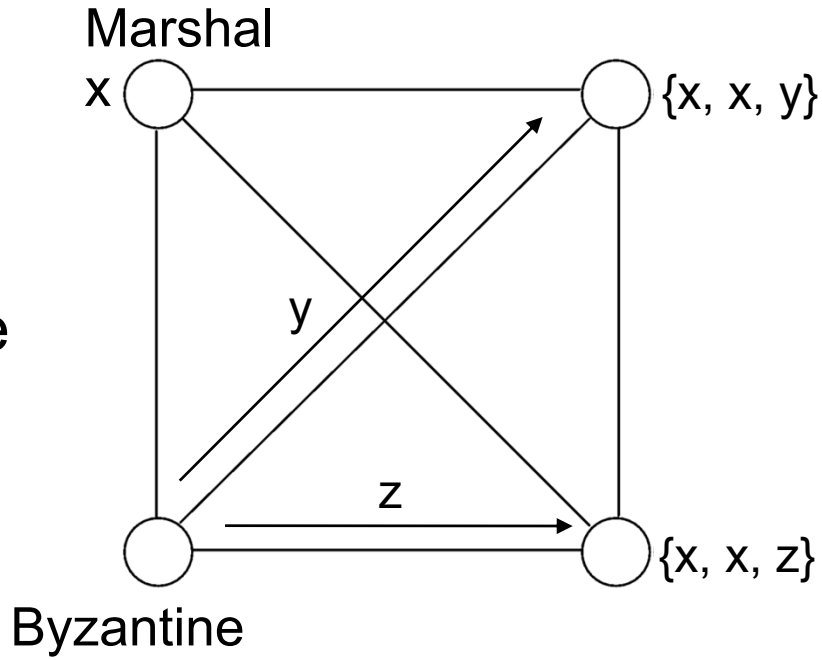
- The marshal sends x. Each node stores x.



Proof of correctness: Validity

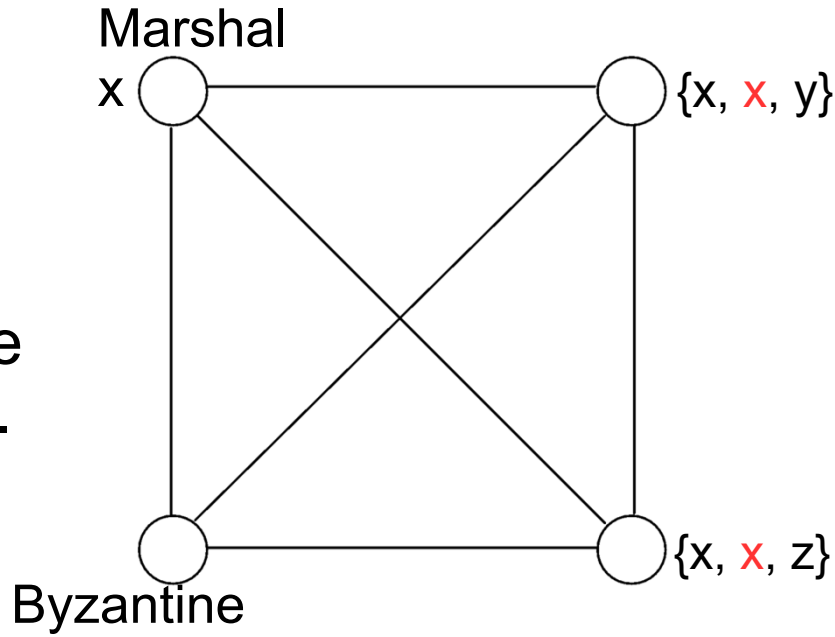
Validity: *If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.*

- The marshal sends x. Each node stores x.
- Then each non-Byzantine node broadcasts x and the Byzantine node sends what it wants (see the figure). → Each node therefore has two times x and the value sent by the Byzantine.



Proof of correctness: Validity

Validity: *If the marshal is non-faulty (correct), the accepted/common decision value of the non-faulty processes is the initial marshal's value.*



- The marshal sends x. Each node stores x.
- Then each non-Byzantine node broadcasts x and the Byzantine node sends what it wants (see the figure).
  → Each node therefore has two times x and the value sent by the Byzantine.

→ Each non-Byzantine node therefore chooses the median value: x

Proof of correctness: Agreement (1)

Agreement: *Any two non-faulty generals never decide/accept different values.*

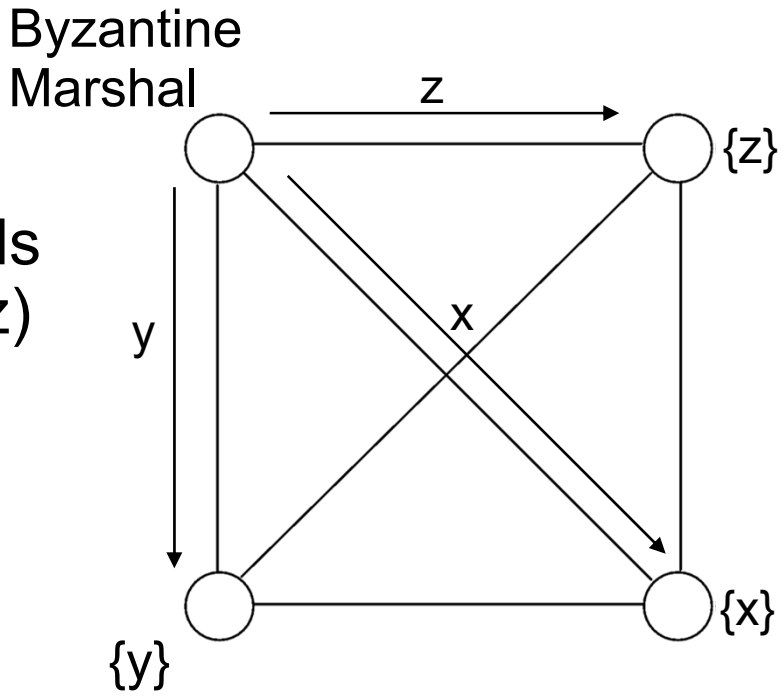
Two cases: either the marshal is Byzantine, or some other is Byzantine.
The case where the marshal is not Byzantine has already been proven for validity condition: the generals decide the same value in this case.

Here, we therefore consider the case in which the marshal is Byzantine.

Proof of correctness: Agreement (2)

Agreement: *Any two non-faulty generals never decide/accept different values.*

Let us consider that the marshal sends possibly different values (x, y and z) to each neighbour.

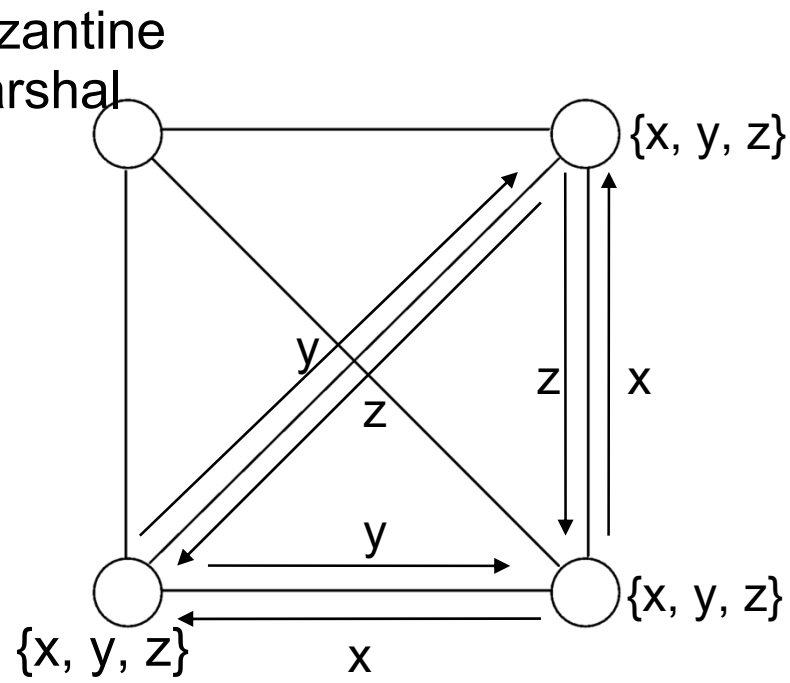


Proof of correctness: Agreement (2)

Agreement: *Any two non-faulty generals never decide/accept different values.*

Let us consider that the marshal sends possibly different values (x, y and z) to each neighbour.

The nodes then exchange the values: each one obtains the values {x, y, z}.



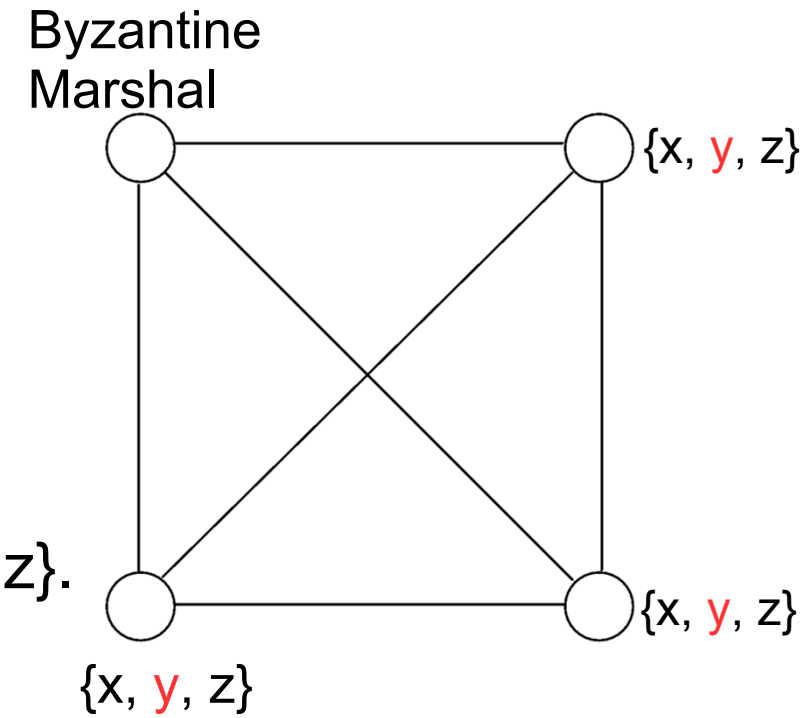
Proof of correctness: Agreement (2)

Agreement: *Any two non-faulty generals never decide/accept different values.*

Let us consider that the marshal sends possibly different values (x, y and z) to each neighbour.

The nodes then exchange the values: each one ends with the values {x, y, z}.

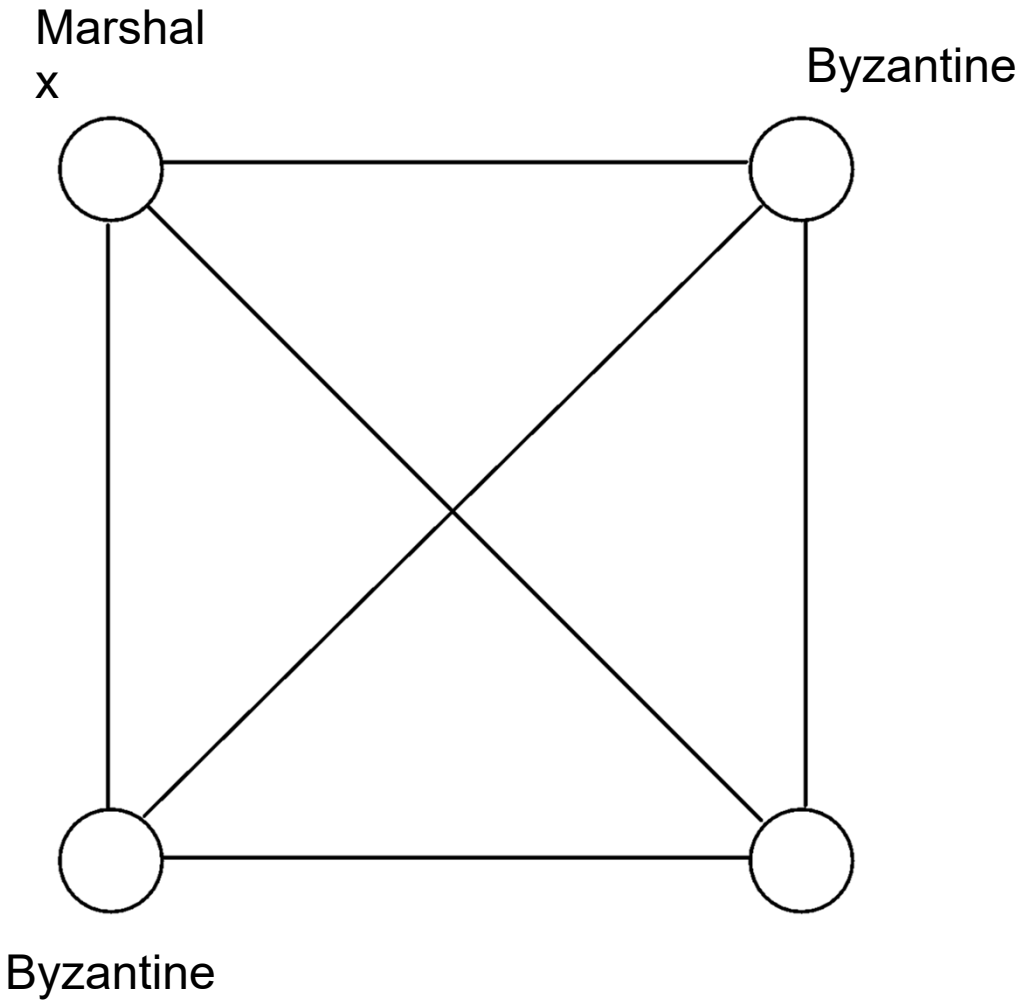
They therefore decide on the same median value.



And for all *n* and *f*?

And for all *n* and *f*?

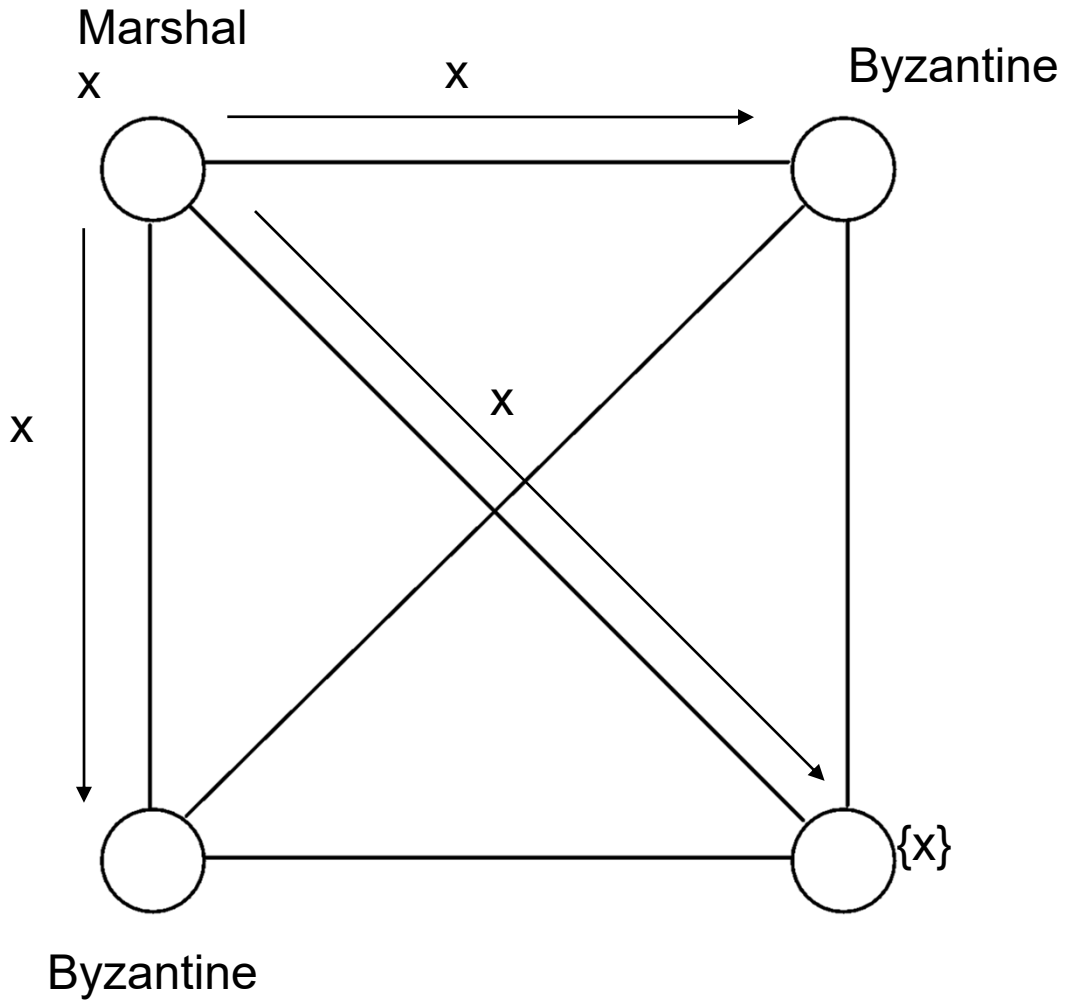
Let us consider a system of 4 nodes with two Byzantines.



And for all *n* and *f*?

Let us consider a system of 4 nodes with two Byzantines.

The marshal sends x.

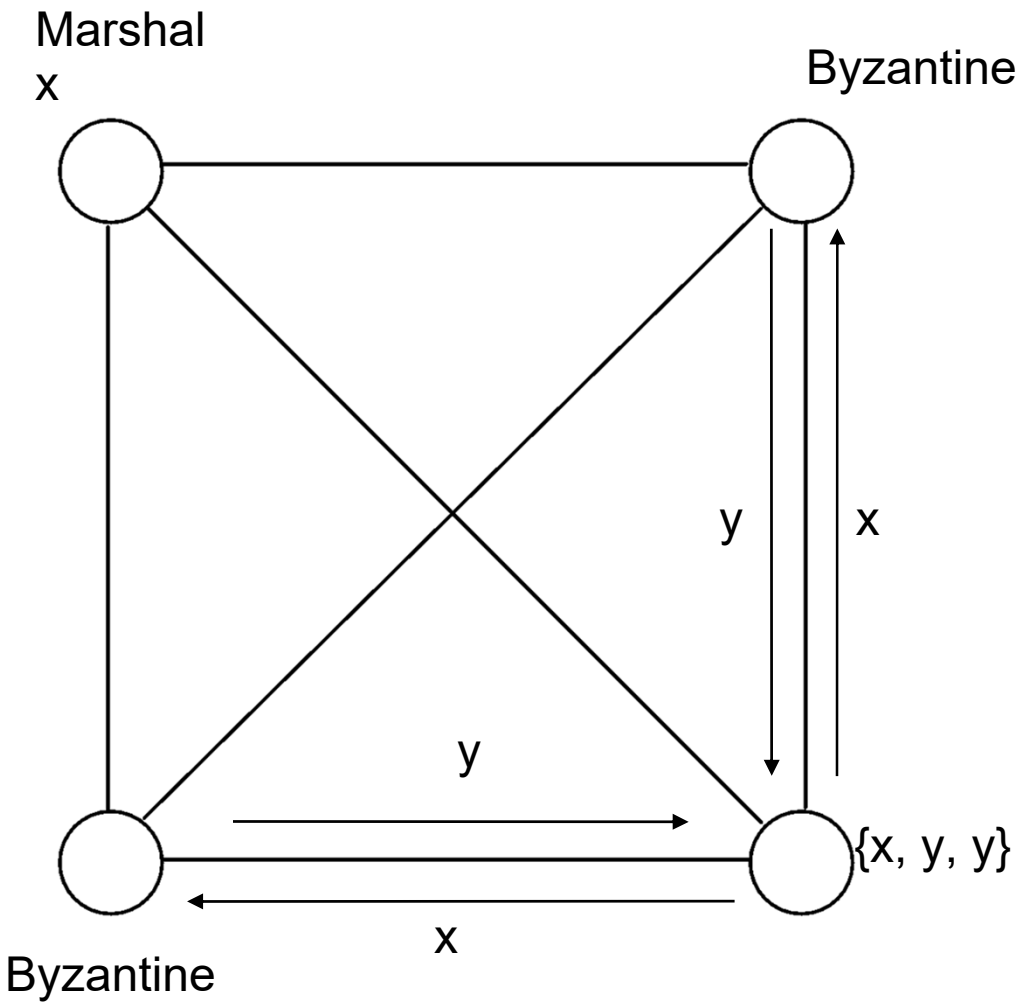


And for all *n* and *f*?

Let us consider a system of 4 nodes with two Byzantines.

The marshal sends x.

The nodes then exchange their values, the Byzantines send what they want.



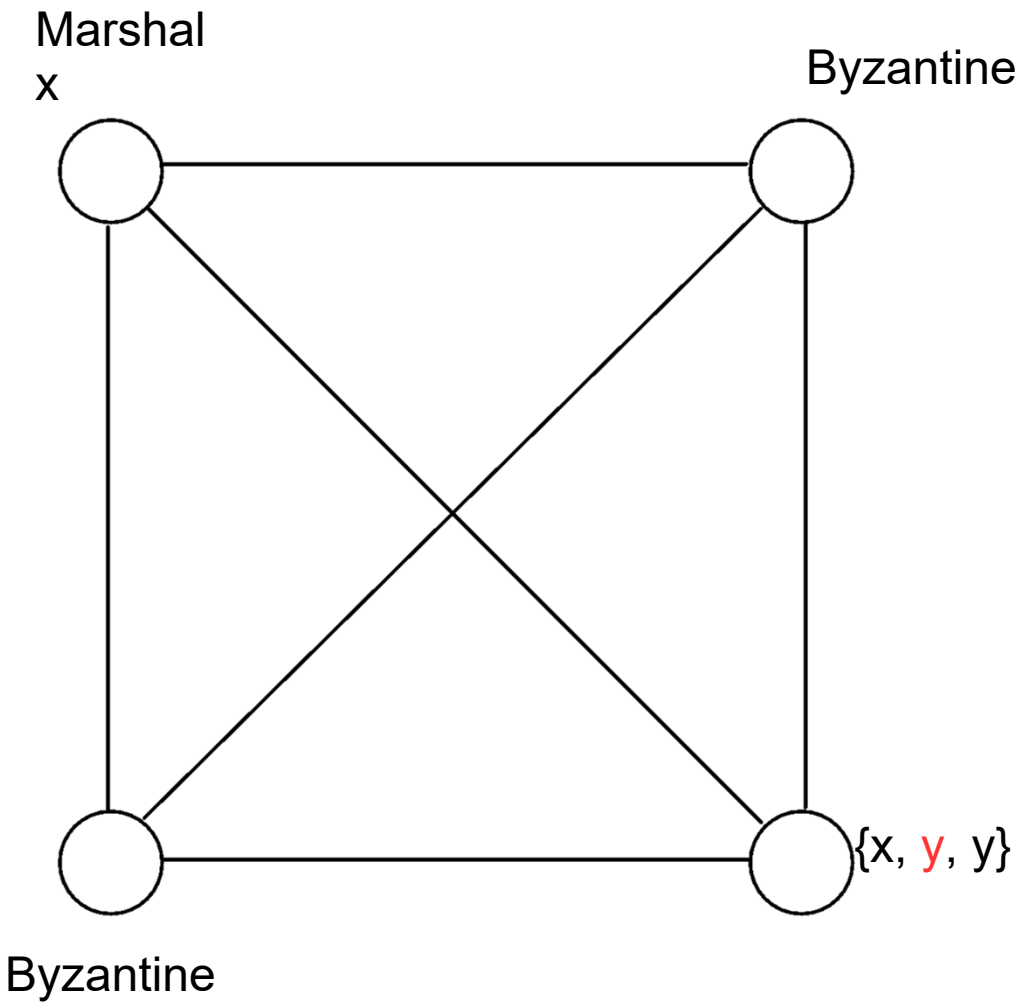
And for all *n* and *f*?

Let us consider a system of 4 nodes with two Byzantines.

The marshal sends x.

The nodes then exchange their values, the Byzantines send what they want.

The non-faulty node chooses y.
→ Validity not respected

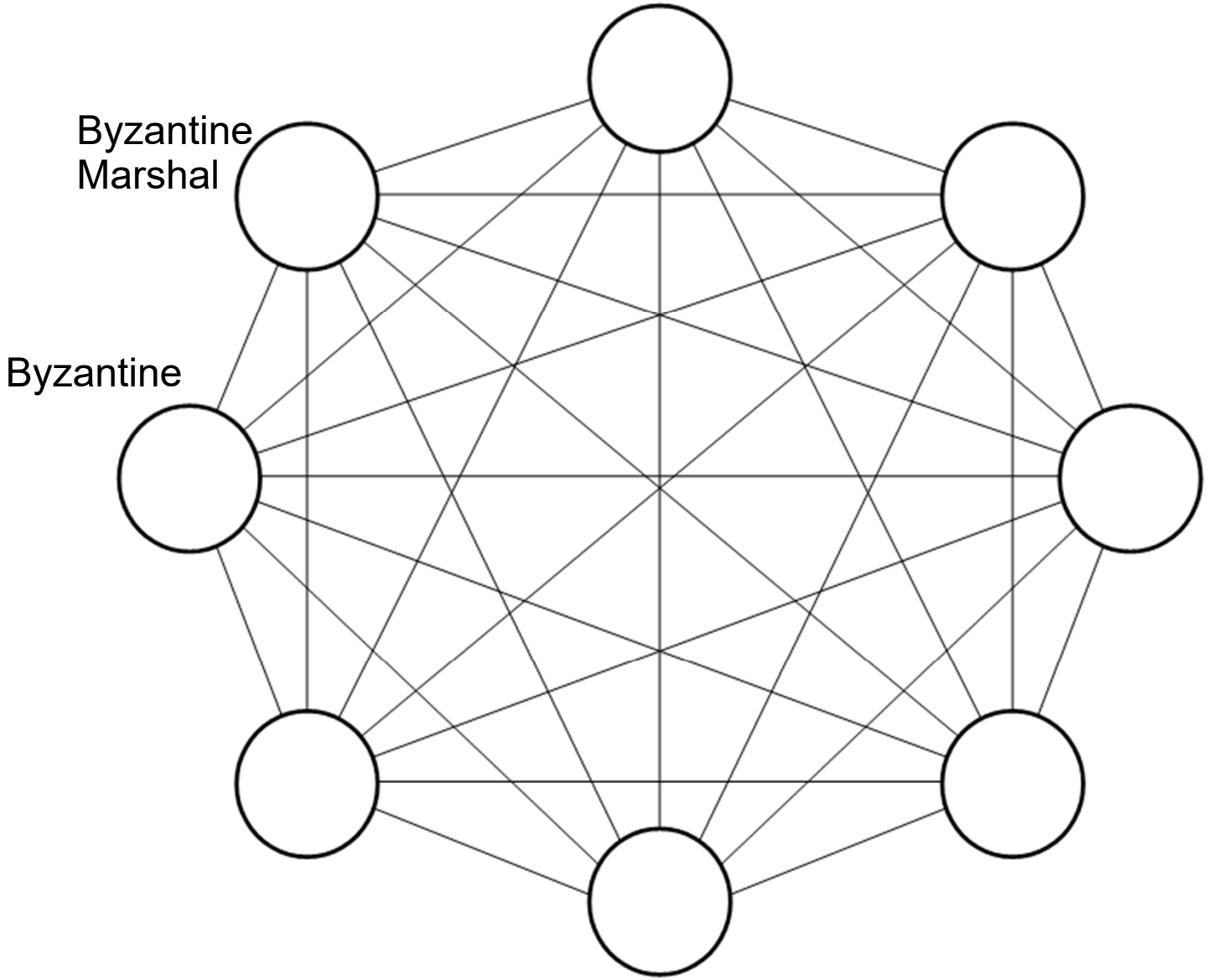


And for all n > 3f?

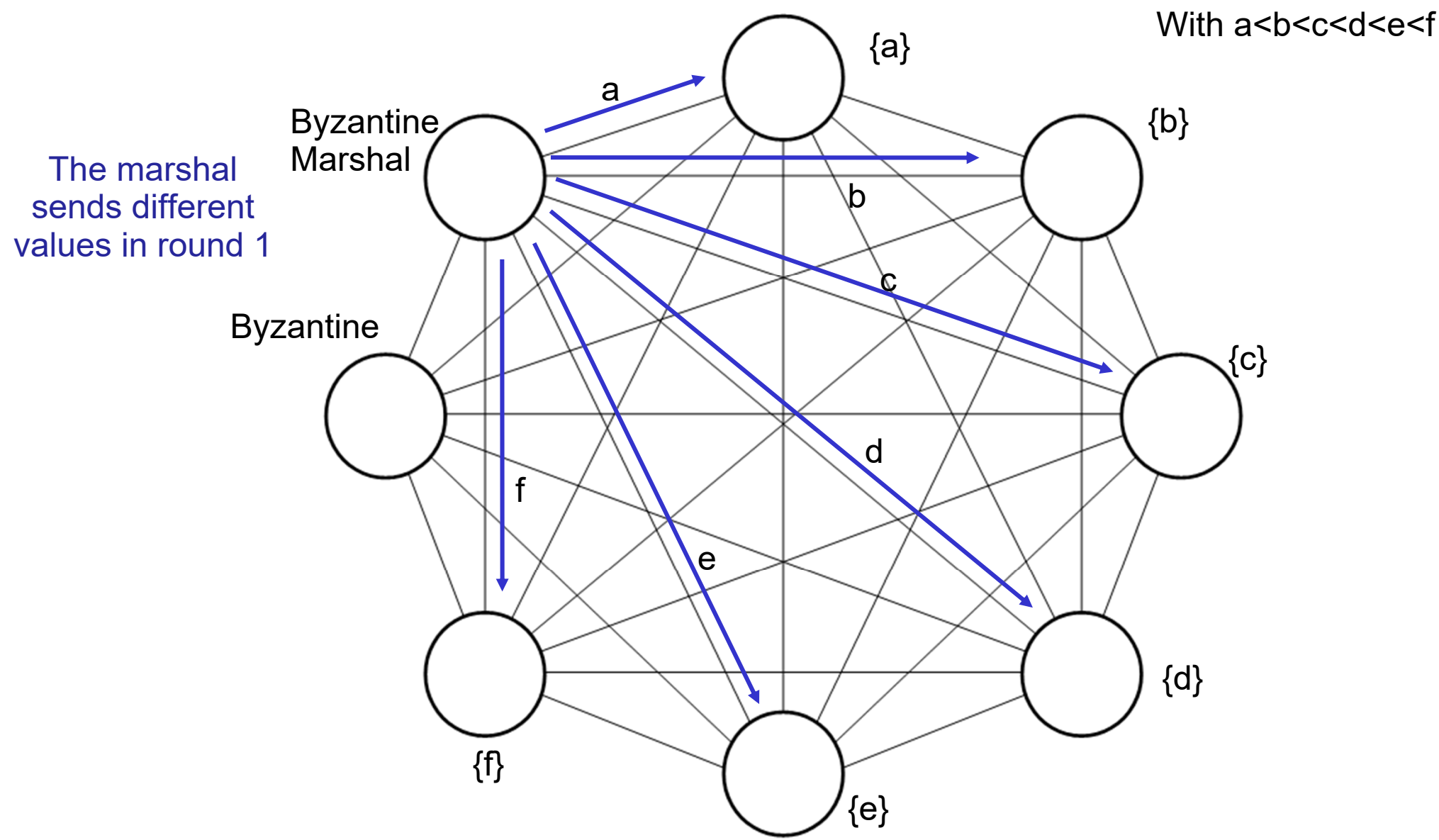
And for all n > 3f?

Let us consider a system with 8 nodes, two of which are Byzantine. The Marshal is Byzantine.

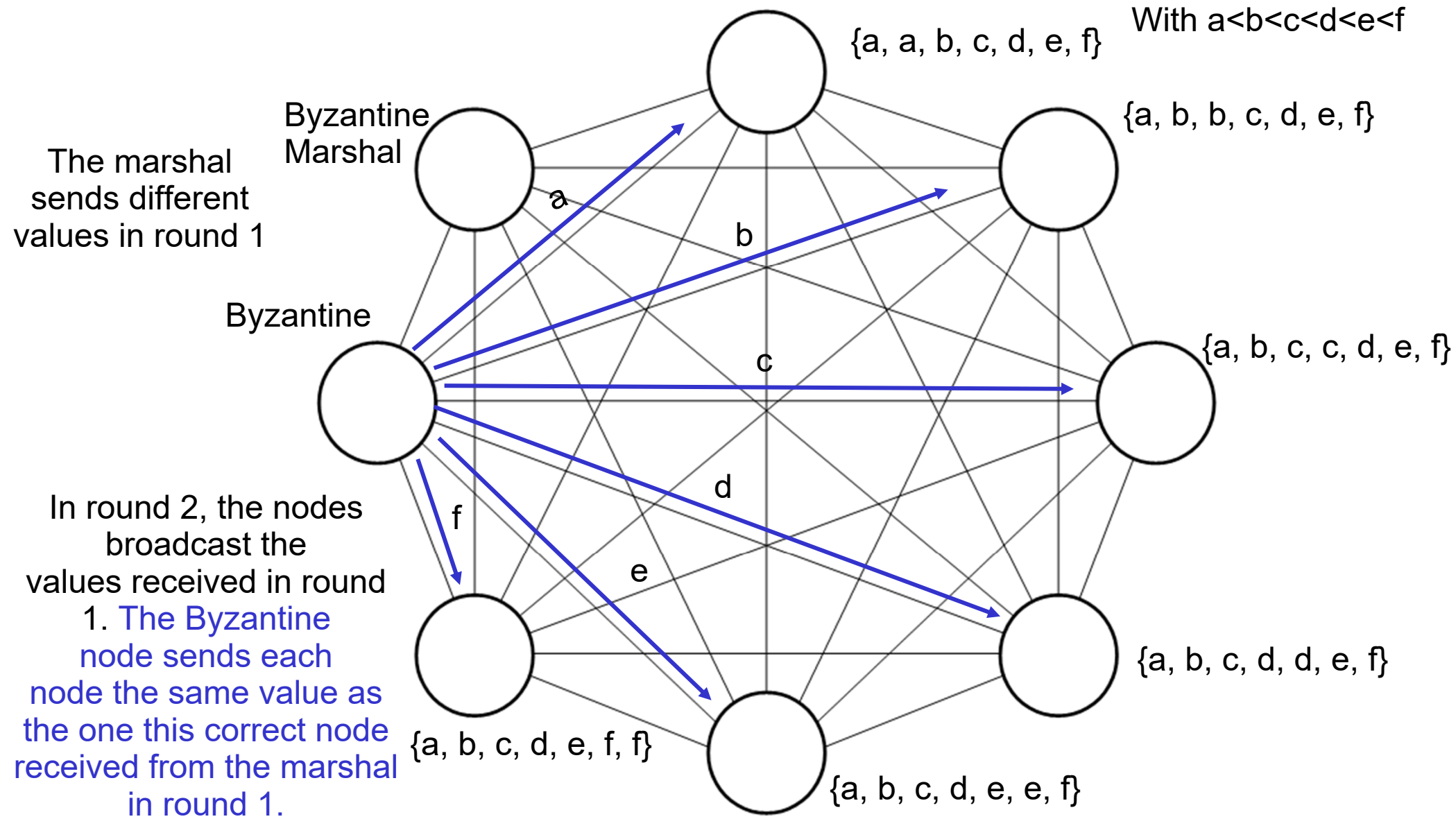
And for all n > 3f?



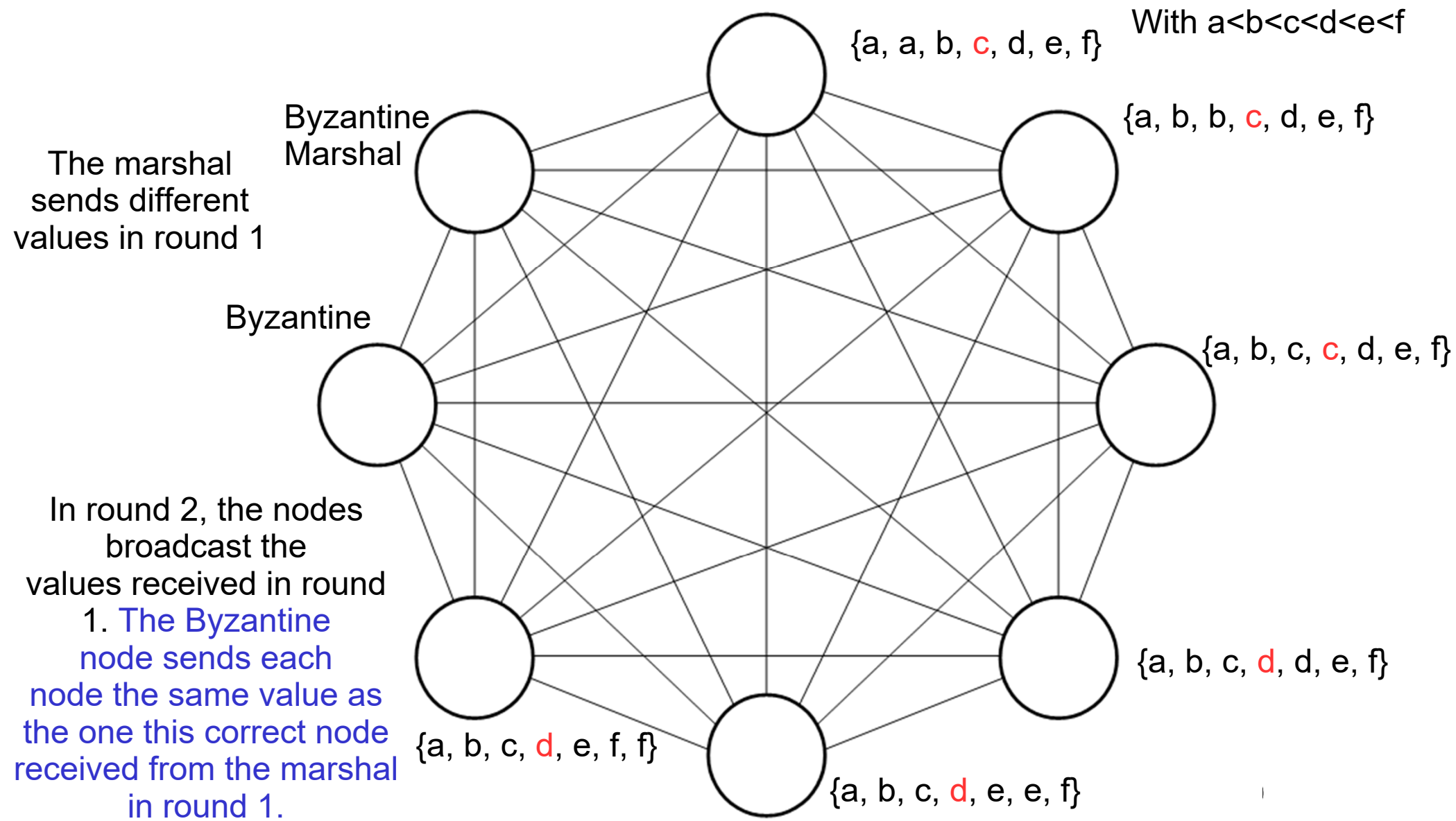
And for all n > 3f?



And for all n > 3f?

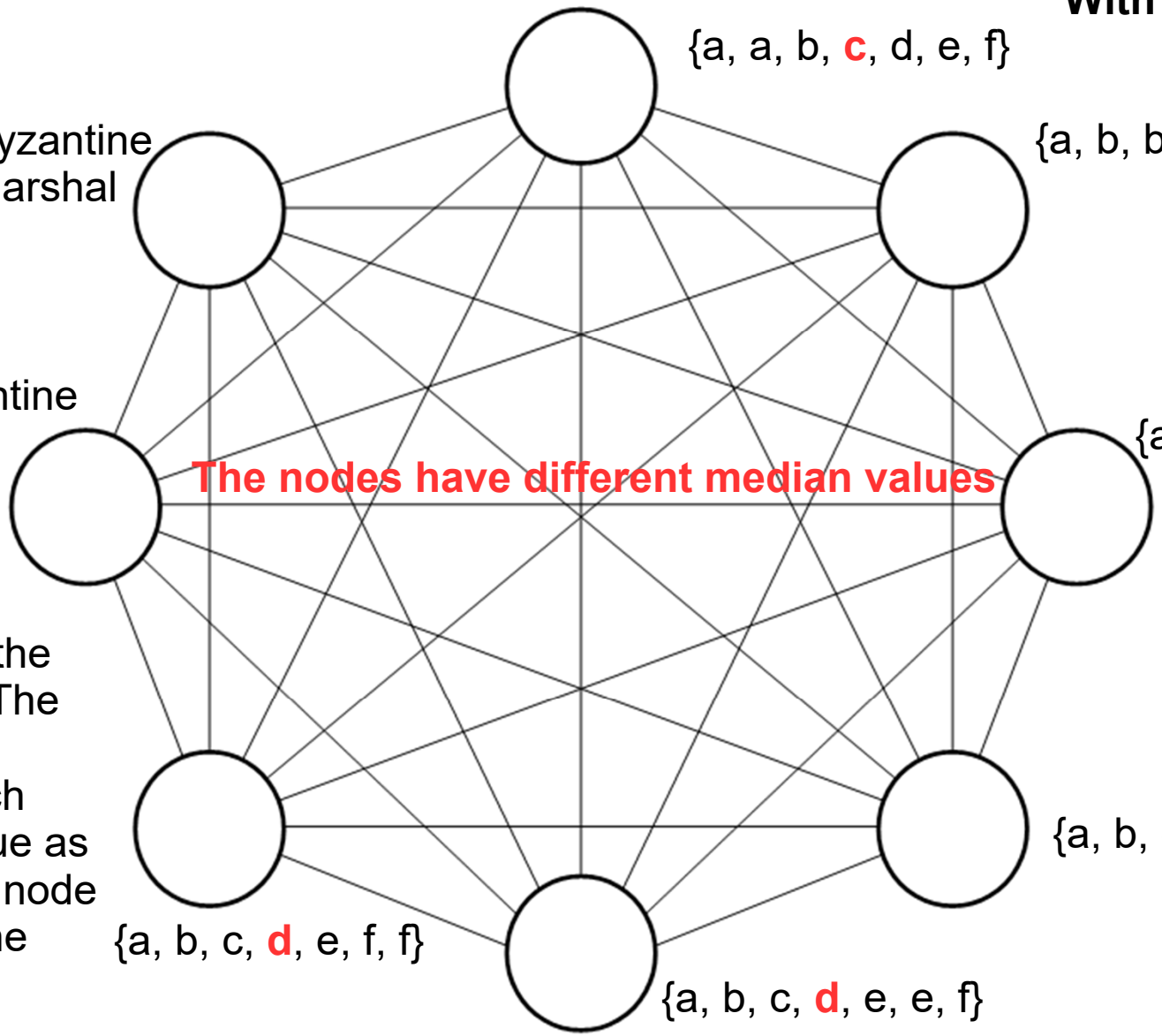


And for all n > 3f?



And for all n > 3f?

With a<b<c<d<e<f



The marshal sends different values

The nodes send the values received. The Byzantine node sends each node the same value as the one this correct node received from the marshal.