30/11/23

Dependable Distributed Systems
Master of Science in Engineering in Computer Science

AA 2023/2024

LECTURE 27: RELIABLE COMMUNICATION IN PRESENCE OF BYZANTINE PROCESSES

Recall: From Perfect Links to Reliable Communication

A link model the capability of a pair processes to exchange messages

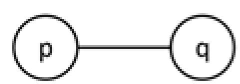
Perfect Point-to-Point Link:

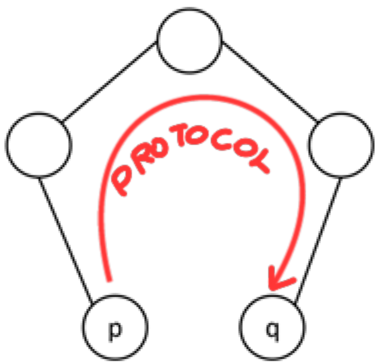
Reliable delivery: If a correct process p sends a message m to a correct process q, then q eventually delivers m.

No duplication

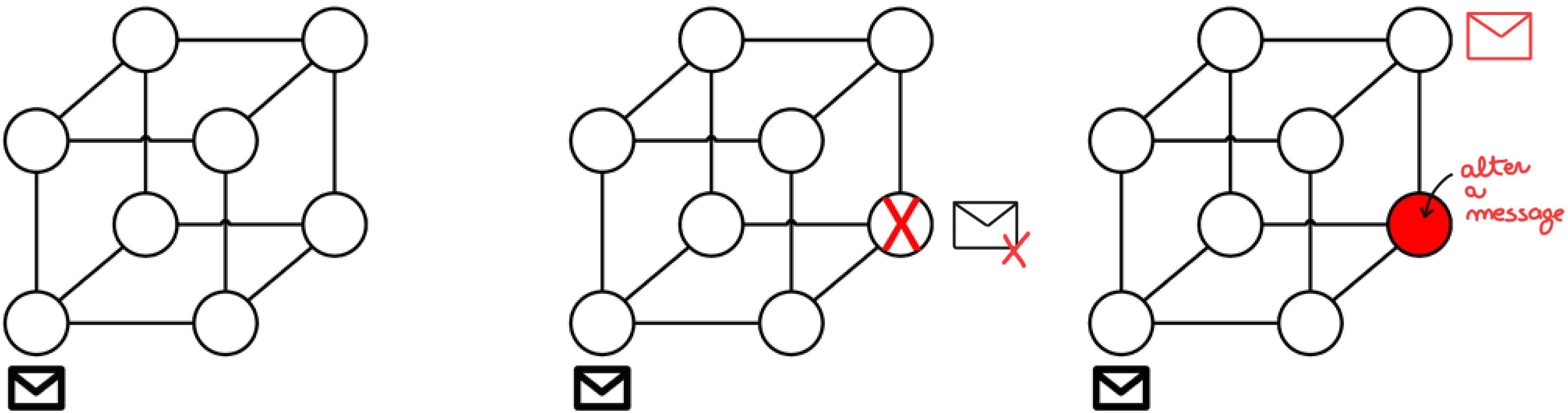
No creation

Can processes p and q exchange messages?





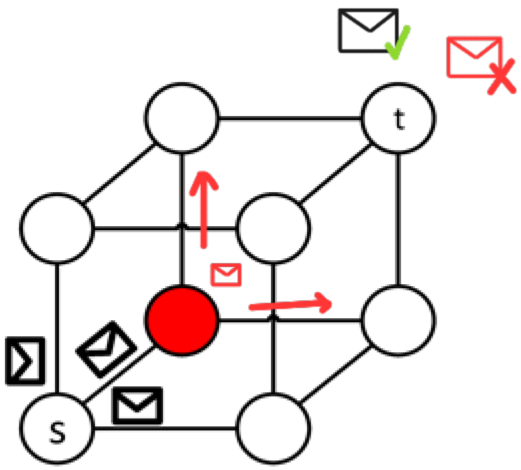
Recall: From Perfect Links to Reliable Communication



Reliable Delivery

No Creation
No Duplication

Reliable Communication Specification



A content can be propagated in the system through several messages

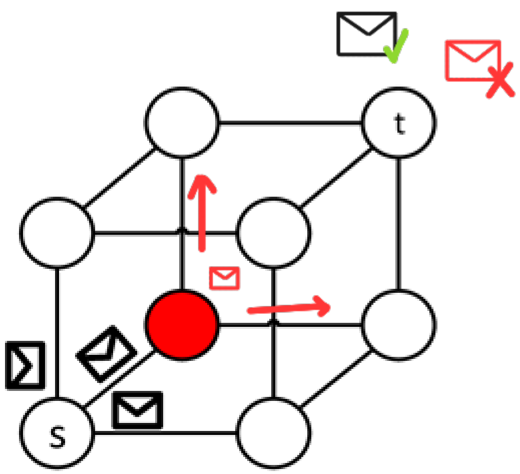
Informally: allow to a pair of processes that are not directly connected by a link to exchange messages (**contents**), guaranteeing their **authorship**, **integrity**, and **delivery.**

Pair of processes: **source** p_s, **target** p_t

Events:

- *Request* < **RC, send | p_t, c** > : Sends a content c to a target process p_t
- *Indication* < **RC, deliver | p_s, c** > : Delivers a content c sent by process p_s

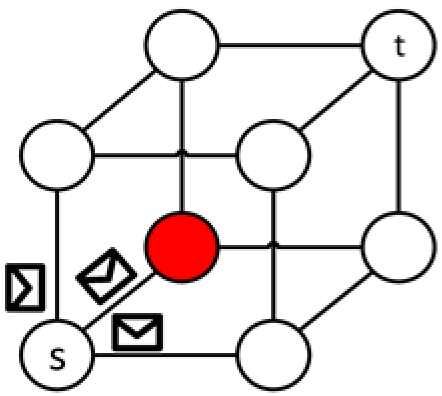
Reliable Communication Specification



- **Safety** = if p_t is a correct process and delivers a **content c** from p_s, then p_s previously sent c (*message integrity and authorship*) the RED m. is not acceptable

- **Liveness** = if p_s is a correct process and sends a **content c** to a correct process p_t , then p_t eventually delivers c from p_s (*message delivery*)

Reliable Communication Specification



The specification can be specialized:

- **one-to-one:** a defined process p_s wants to reliably communicate with a specific target process p_t
- **any-to-any:** any process wants to reliably communicate with any other process

Reliable Communication

What kind of faults may occur?

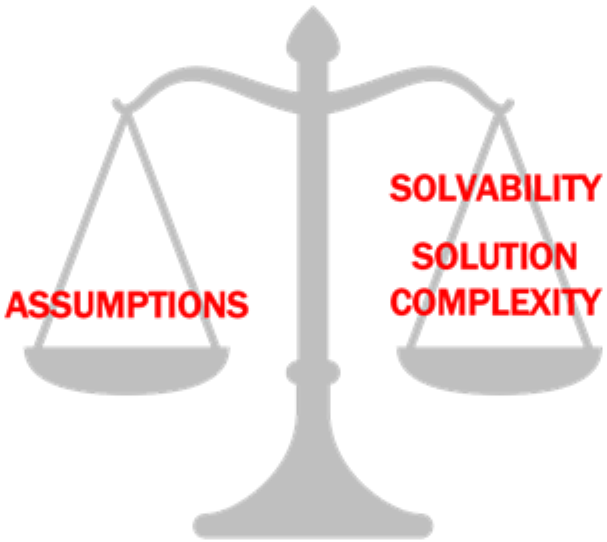
How many faults may occur?

How faults are distributed?

Which facilities are available?

What knowledge the processes have about the system?

…



Menger Theorem

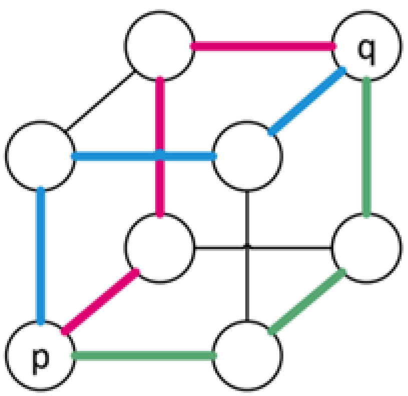
Menger Theorem - ***Vertex Cut*** *VS* ***Disjoint Paths***: Let $G = (V, E)$ be a graph and $p, q \subseteq V$. Then the minimum number of vertexes separating p from q in G is equal to the maximum number of disjoint $p - q$ paths in G.

https://en.wikipedia.org/wiki/Menger%27s_theorem

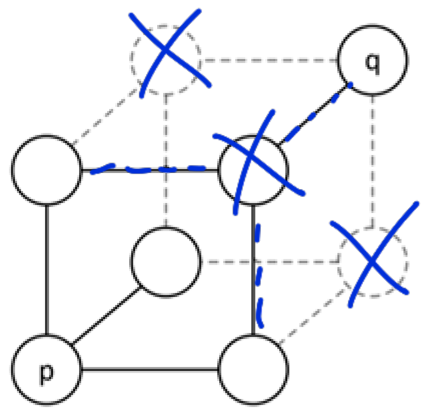
IF I REMOVE 3 NODE P AND Q ARE NOT LINKED ANYMORE

↓

3 CUT



Disjoint Paths



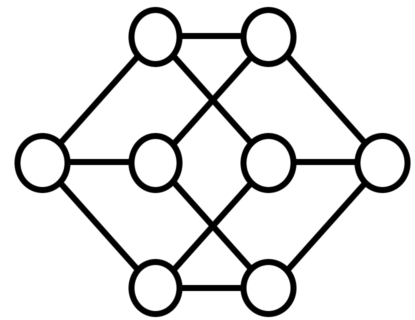
Min Cut

NOTE: **the min-cut can be computed with a polynomial algorithm** in the size of the graph

Globally Bounded Fault Model - Crashes

System model:

- **at most f processes can be faulty**, crash faults
- n processes
- **perfect point-to-point links**

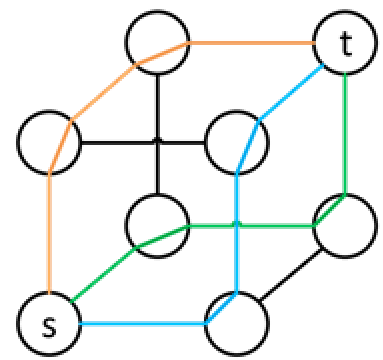


Correctness Conditions:

2

- *one-to-one*: **At least f+1 node-disjoint paths** must exist between the source and the target processes
- *any-to-any*: **node-connectivity k > f+1**

minimum number of node that have to be removed to disconnect the network

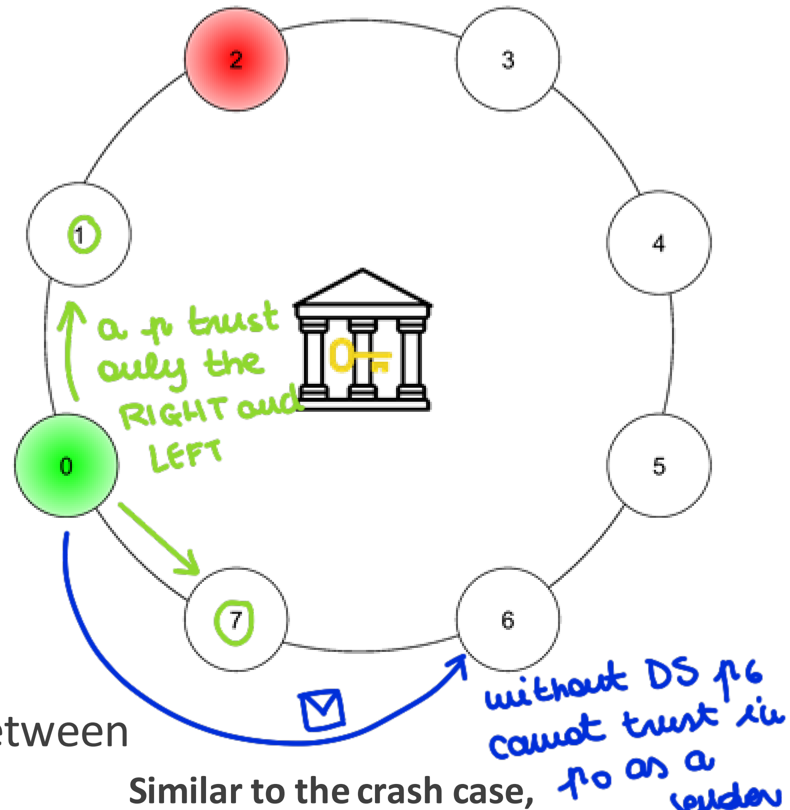


Globally Bounded Fault Model – Byzantine Faults, Authenticated Messages

System model:

- **at most f processes can be faulty**, Byzantine faults
- n processes
- **perfect point-to-point links**
- **digitally signed messages – authenticated messages**

(every process can sign ONLY the messages it generates and verify the signatures generated by every process)



Correctness Conditions (same as the crash case):

one-to-one: **At least f+1 node-disjoint paths** must exist between the source and the target processes

any-to-any: **node-connectivity k > f+1**

Similar to the crash case, contents attached with a digital signature, only contents with a valid digital signature are considered

Globally Bounded Fault Model – Byzantine Faults, Authenticated Links

System model:

- at most f processes can be faulty, Byzantine faults
- n processes
- authenticated perfect point-to-point links

  authenticated links
- processes know the topology of the overlay network

weaker respect to DS

_> every process can (deterministically) compute node-disjoint paths between all pairs of processes

Globally Bounded Fault Model – Byzantine Faults, Authenticated Links

DEST content

upon DolevR_send(t, c) **do**

  for $\pi \in \Pi_{i,t}$ **do** path

    send($\langle i, t, c, \pi \rangle$, $\pi[1]$) 2° node of the path

pp2p, deliver

SOURCE

upon receive($\langle s, t, c, \pi \rangle$, j) **do**

  if $\pi \in \Pi_{s,t}$, $\exists m \in \mathbb{N}^0$, $\pi[m-1] = j$, $\pi[m] = i$ **then**

    if $|\pi| = m$ **then** → if I'm the final DEST

      $Paths_{\langle s,c \rangle} \leftarrow Paths_{\langle s,c \rangle} \cup \pi$

    else Paths: data structure

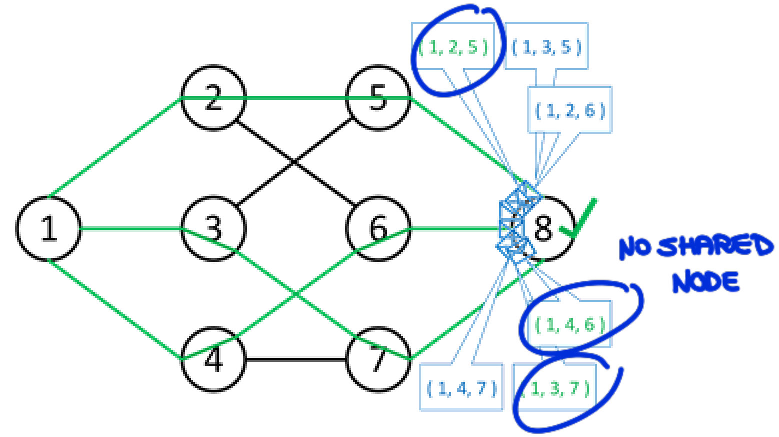
      send($\langle s, t, c, \pi \rangle$, $\pi[m+1]$)

here m is an index

upon $|Paths_{\langle s,c \rangle}| > f$ **do**

  DolevU_deliver($\langle s, c \rangle$)

Optimal message complexity: O(n)
Optimal delivery complexity: O(f)



Correctness: node-connectivity > 2f

Question: why 2f is enough?

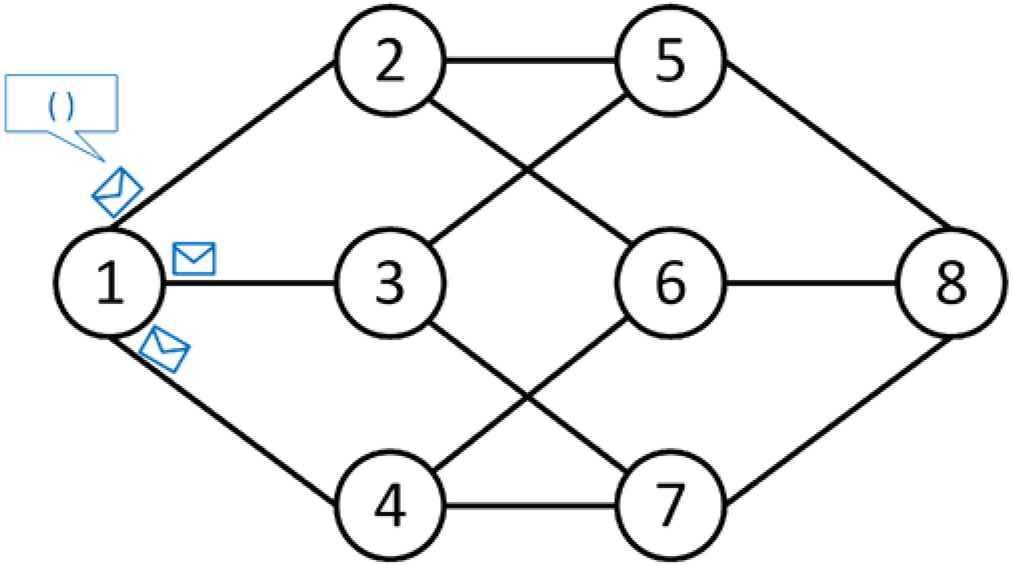
Globally Bounded Fault Model – **Byzantine** Faults, **Authenticated Links**

System model:

- **at most f processes can be faulty**, Byzantine faults
- n processes
- **<u>authenticated</u> perfect point-to-point links**
- ~~processes know the topology of the communication~~

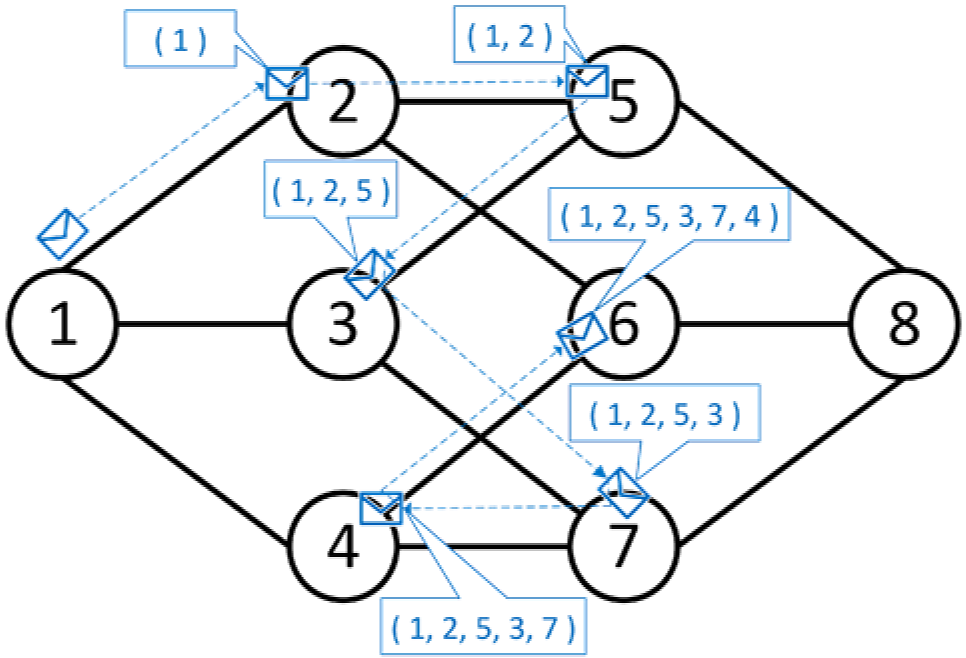
_> Flooding

DolevU – Propagation

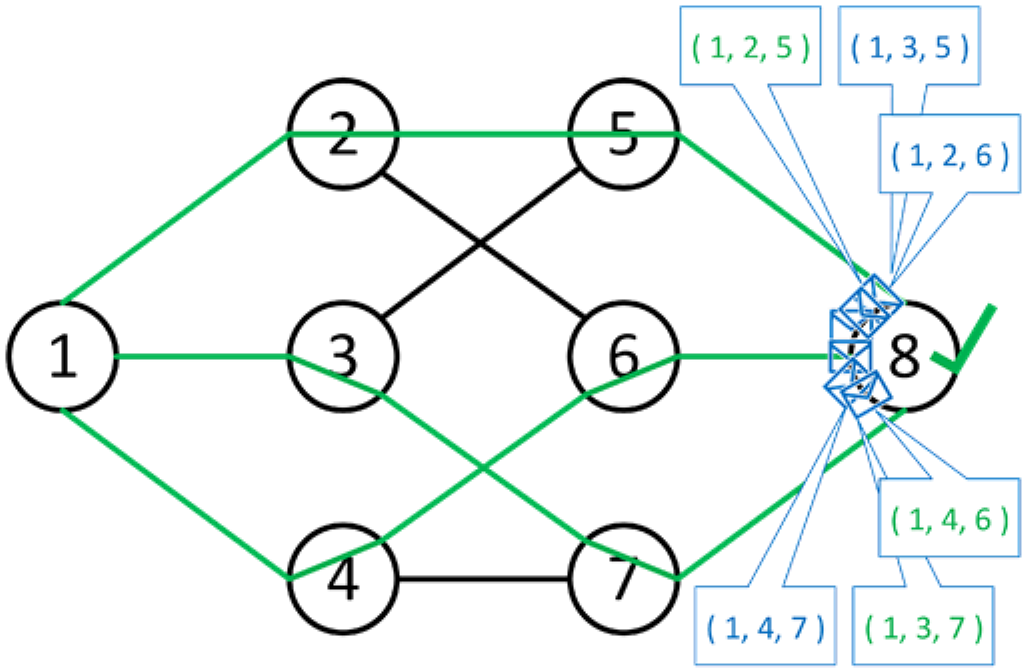


Note: only part of the messages exchanged are shown

DolevU - Issue



DolevU - Delivery



Globally Bounded Fault Model – Byzantine Faults, Authenticated Links

here * stands for every process (target)

upon DolevU_send(c) **do**
  for $j \in \Gamma(i)$ **do**
    send($\langle i, *, c, \emptyset \rangle, j$)

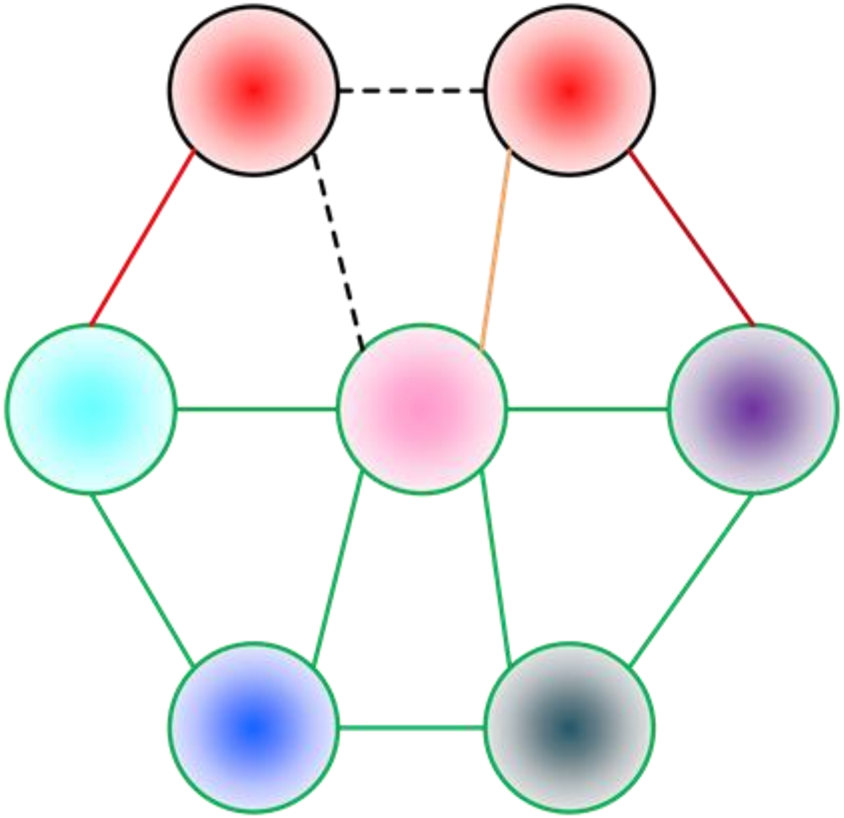
upon receive($\langle s, *, c, path \rangle, j$) **do**
  $path \leftarrow path \cup \{j\}$
  $Paths_{\langle s,c \rangle} \leftarrow Paths_{\langle s,c \rangle} \cup \{path\}$
  for $j \in \Gamma(i)$ **do**
    if $j \notin path$ **then**
      send($\langle s, c, path \rangle, j$)

upon max_disjoint_paths($Paths_{\langle s,c \rangle}$) $> f$ **do**
  DolevU_deliver($\langle s, c \rangle$)

message complexity: O(n!)
delivery complexity: NP Complete problem!

Correctness: node-connectivity > 2f

(EXTRA) Byzantine Tolerant Topology Reconstruction



It is possible to define a RC protocol
with optimal message complexity
and delivery complexity if stronger assumptions
(about the links or about the node-connectivity
of the network) are assumed

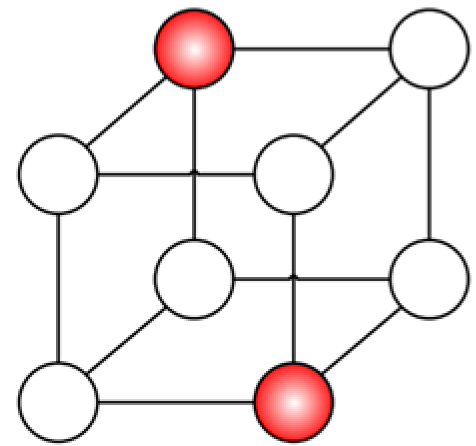
Locally Bounded Fault Model – **Byzantine** Faults, **Authenticated Links**

System model:

- **at most f processes can be faulty <u>in the neighborhood of every node</u>**, Byzantine faults

- n processes

- **authenticated perfect point-to-point links**



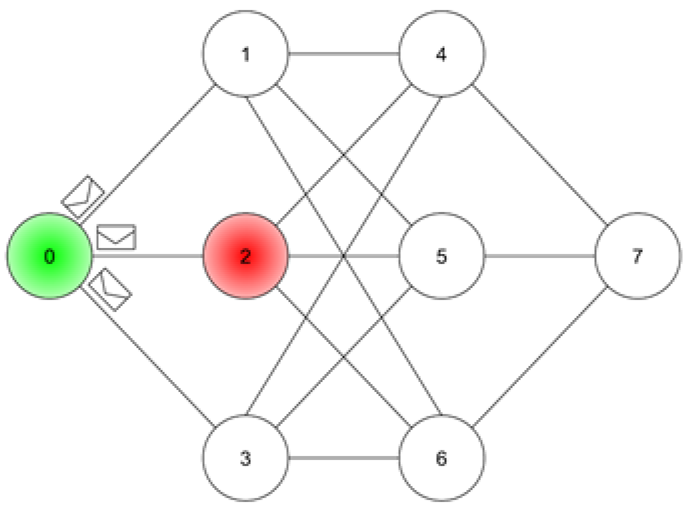
1-local distribution
2-global distribution

Question: can you solve RC on that topology?

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)

Message Format



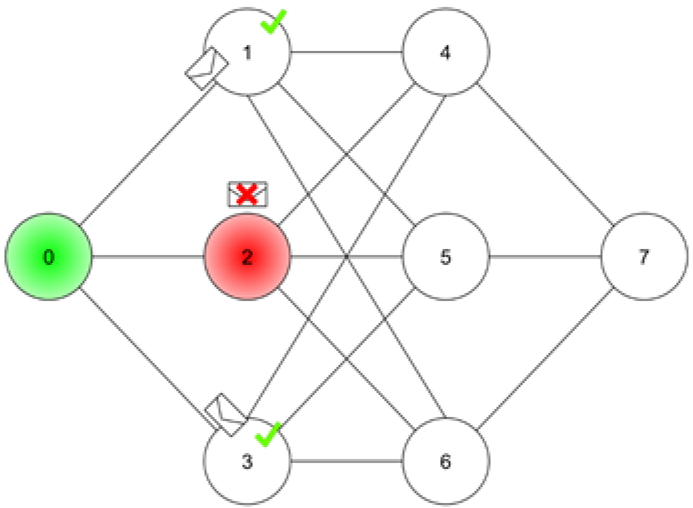
$f = 1$

- **the source broadcasts the message**;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)

Message Format



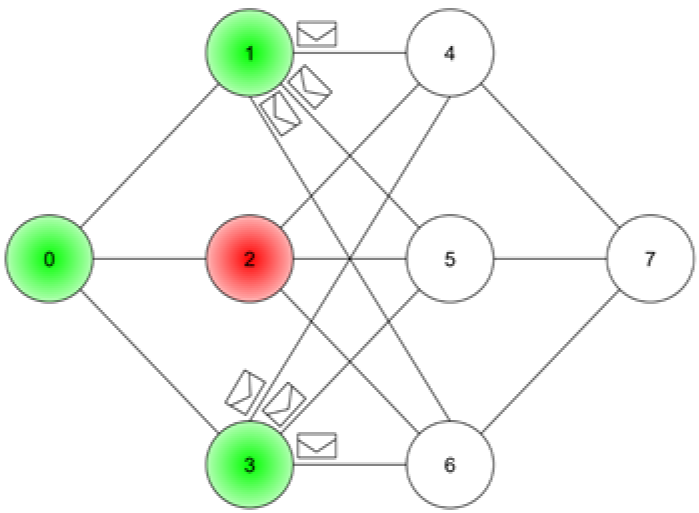
f = 1

- the source broadcasts the message;
- **a neighbor of the source directly accepts and relays the message**;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)

Message Format



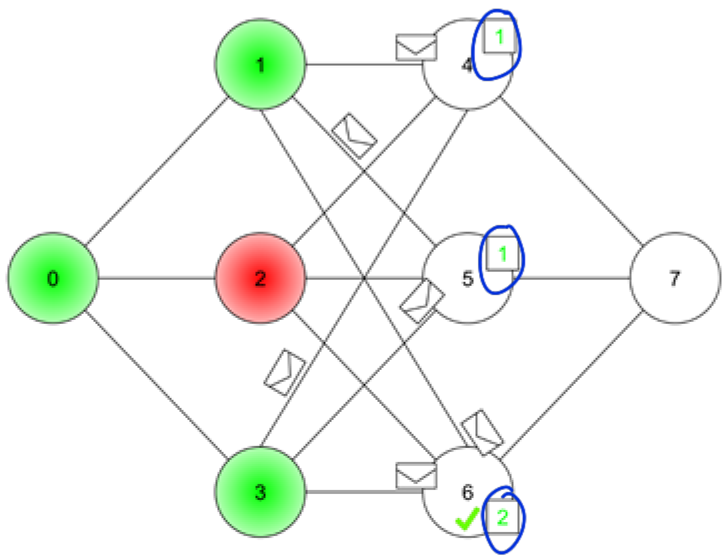
f = 1

- the source broadcasts the message;
- **a neighbor of the source directly accepts and relays the message**;
- a process that receives the same message from $f+1$ distinct neighbors accepts and relays the message.

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)

Message Format



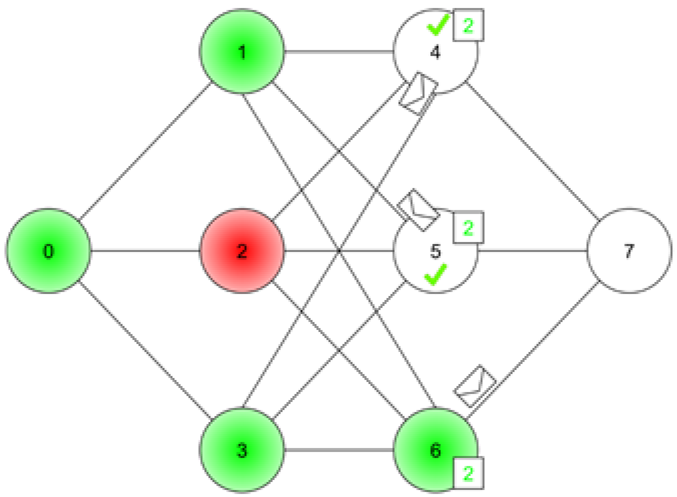
f = 1

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- **a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.**

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)

Message Format

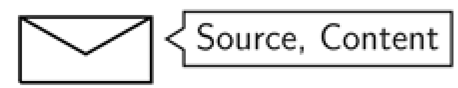


f = 1

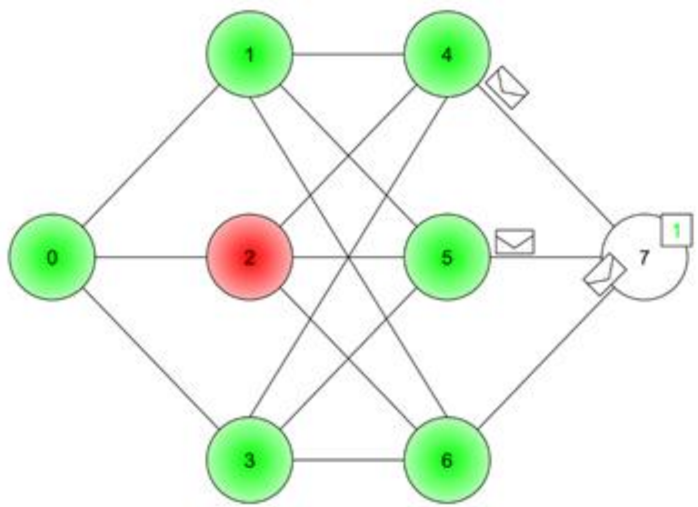
- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- **a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.**

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

Certified Propagation Algorithm (CPA)



Message Format



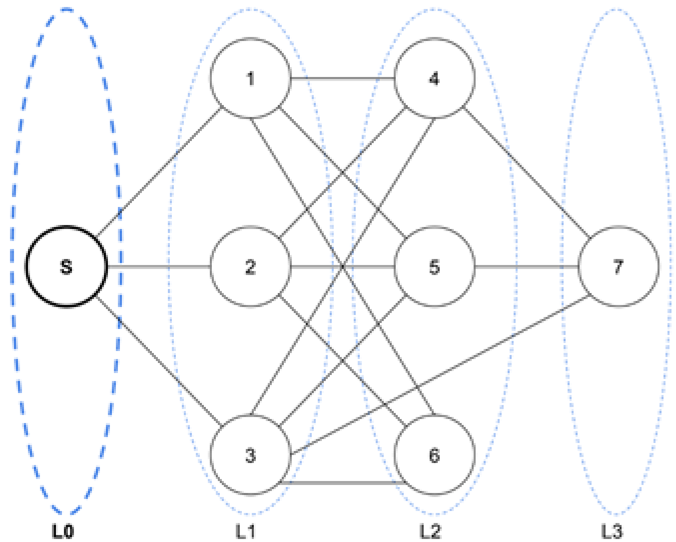
f = 1

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- **a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.**

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

CPA Correctness (from a specific source) – Minimum k-level ordering (MKLO)

MKLO = Partition of the nodes in levels



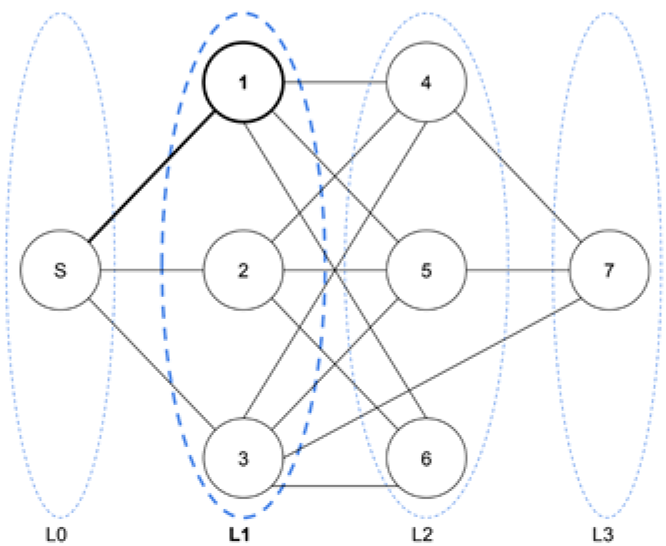
$k = 3$

- **The source is placed in** $L0$;

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

CPA Correctness (from a specific source) – Minimum k-level ordering (MKLO)

MKLO = Partition of the nodes in levels



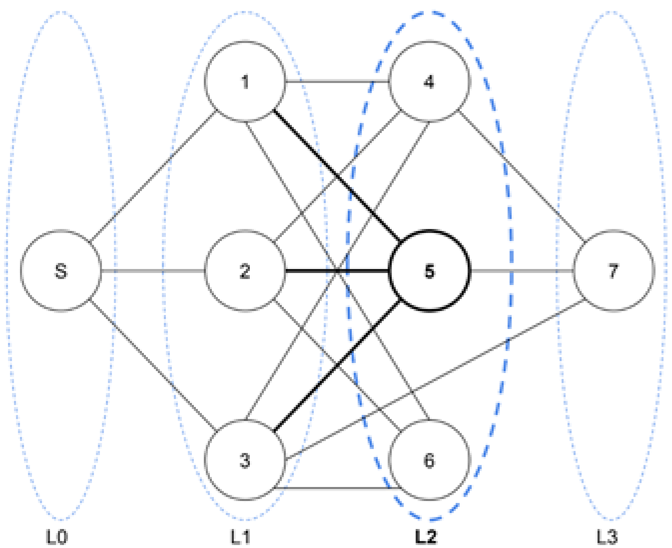
k = 3

- The source is placed in $L0$;
- **The neighbors of the source are placed in level $L1$;**

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

CPA Correctness (from a specific source) – Minimum k-level ordering (MKLO)

MKLO = Partition of the nodes in levels



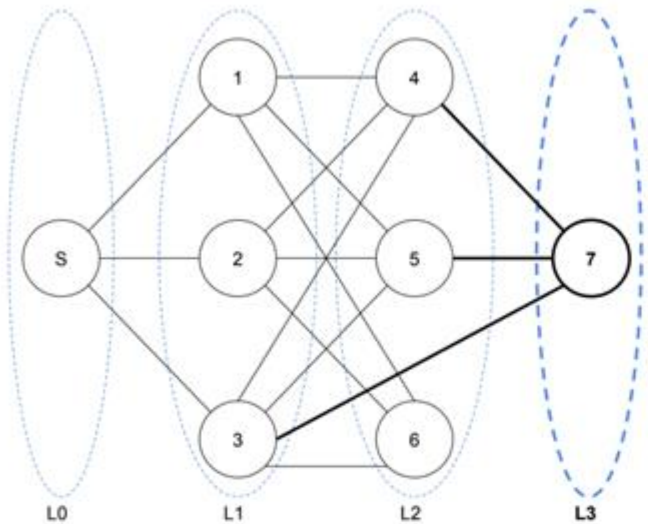
k = 3

- The source is placed in $L0$;
- The neighbors of the source are placed in level $L1$;
- **Any other node is places in the first level such that it has at least k neighbors in the previous levels.**

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

CPA Correctness (from a specific source) – Minimum k-level ordering (MKLO)

MKLO = Partition of the nodes in levels



$k = 3$

- The source is placed in $L0$;
- The neighbors of the source are placed in level $L1$;
- **Any other node is places in the first level such that it has at least k neighbors in the previous levels.**

NOTE: you must include all nodes in MKLO

Locally Bounded Fault Model – Byzantine Faults, Authenticated Links

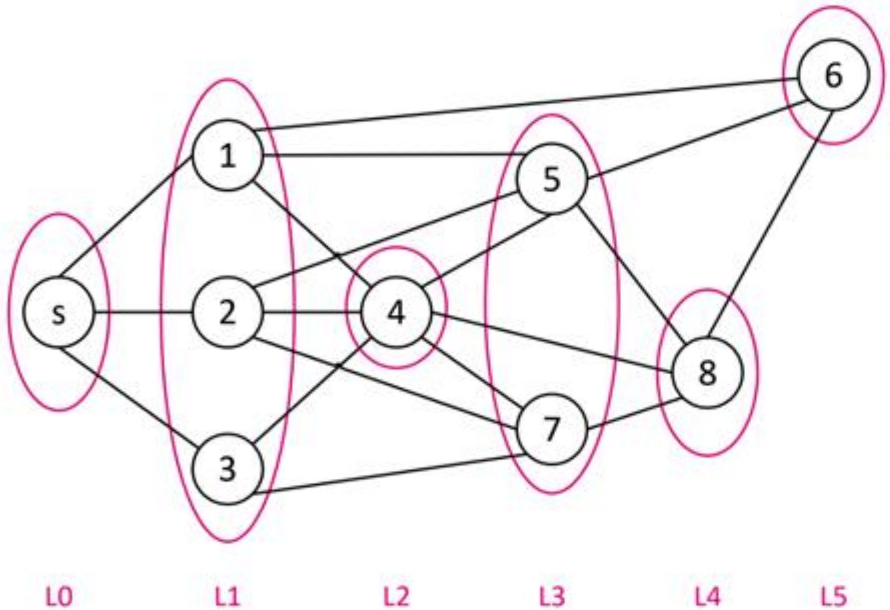
CPA Correctness (from a specific source) – Minimum k-level ordering (MKLO)

Necessary condition: MKLO with $k = f{+}1$

Sufficient condition: MKLO with $k = 2f{+}1$

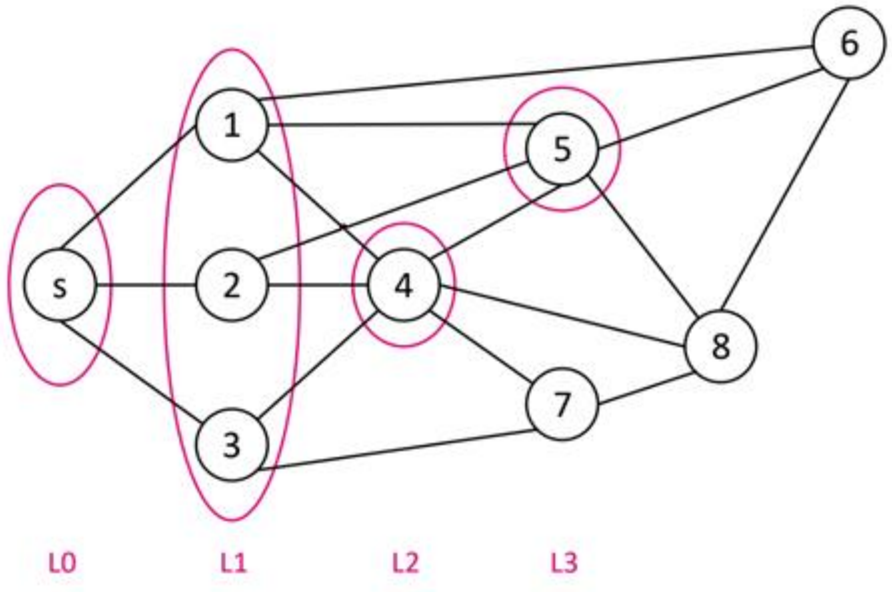
Strict condition: MKLO with $k = f{+}1$ removing any possible placement of the Byzantine processes (NP-Complete Problem)

MKLO Example



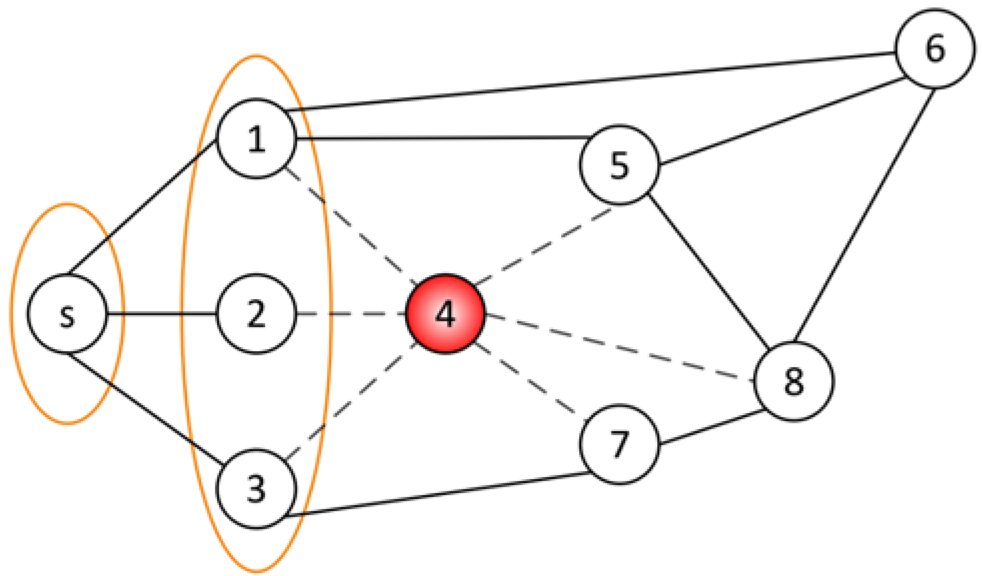
MKLO with K=3

MKLO Example



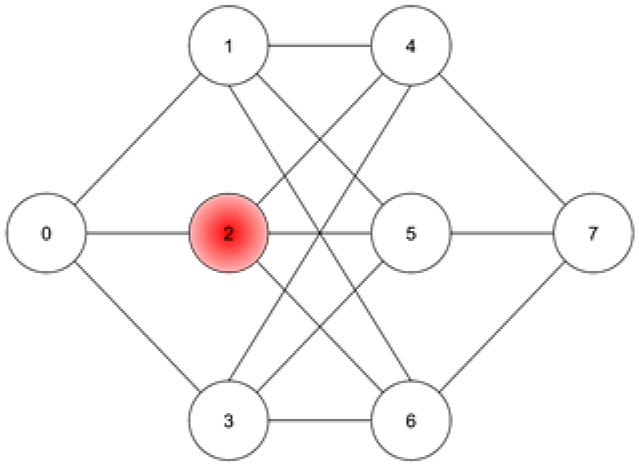
MKLO with K=3 is not defined
MKLO with K=2 is defined for every 1-local removal

MKLO Example



MKLO with K=2 is not defined removing node 4

CPA Recap



CPA: $O(n^2)$ **messages, delivery** $O(f)$

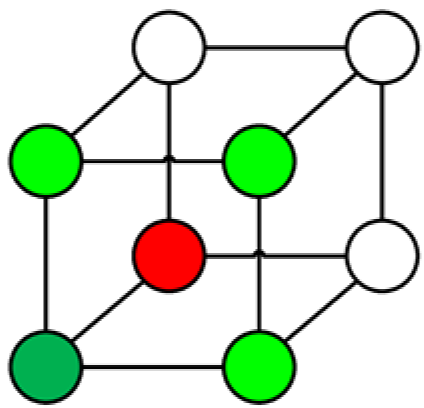
Exact Solvability: NP-Complete

Globally vs Locally Bounded

f-global distribution => f-local distribution (for the same value of f)

The vice-versa is not true (1-local may imply >1-global)

_> If the topology is unknown, you may attempt to use CPA to solve RC in the globally bounded failure model, but a «more dense» topology is required



1-local/1-global distribution

BRB in General Networks

Question: it is possible for a faulty source using an ByzRC primitive to send different contents to distinct processes?

System model:
- n processes
- at most f Byzantine faulty processes
- authenticated links
- not fully connected topology

How you can solve the problem?

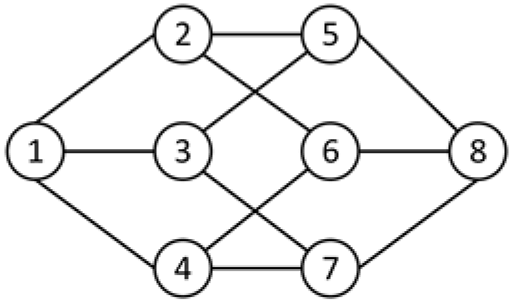
What are the correctness conditions?

Why reliable communication is usefull?

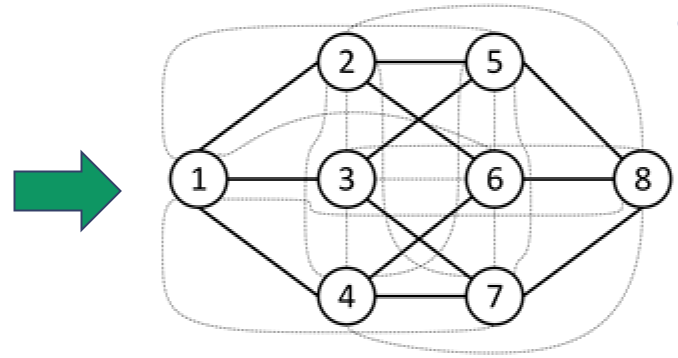
Simulate a **complete communication network**

_> use all the solutions defined for fully-connected distributed systems

Create virtual PP2PL in order to exange m between p that are not directly connected



Communication Network (made of P2P-link)



Overlay Network

References

- Danny Dolev. *Unanimity in an unknown and unreliable environment* https://doi.org/10.1109/SFCS.1981.53

- Andrzej Pelc and David Peleg. *Broadcasting with locally bounded byzantine faults* https://doi.org/10.1016/j.ipl.2004.10.007

- Chris Litsas, Aris Pagourtzis, and Dimitris Sakavalas. *A graph parameter that matches the resilience of the certified propagation algorithm* https://doi.org/10.1007/978-3-642-39247-4_23.

- Giovanni Farina. *Tractable Reliable Communication in Compromised Networks* https://tel.archives-ouvertes.fr/tel-03118108