Topics in Distributed Algorithms: On TDMA for Ad Hoc Networks and Coded Atomic Storage Algorithms

Thomas Petig

Chalmers University of Technology

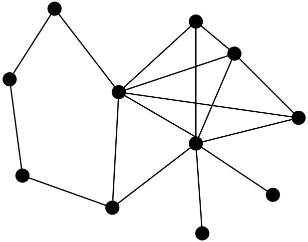
Distributed Computing and Systems
Chalmers university of technology

October 19, 2015

Distributed Algorithms

- Computational unit: node
- Communication channels

We model the system as a graph, where
nodes → vertices and
communication channels → edges.



Fault-tolerance

What faults are considered:

- Transient fault: data gets corrupted
  - $\rightsquigarrow$ Self-stabilization
- (Semi) Byzantine failure: node behaves arbitrarily
  - $\rightsquigarrow$ Erasure codes

We are going to discuss these publications:

- Self-stabilizing TDMA Algorithms for Wireless Ad-hoc Networks without External Reference.
- Robust and Private Distributed Shared Atomic Memory in Message Passing Networks.

Self-stabilizing TDMA Algorithms for Wireless Ad-hoc Networks without External Reference

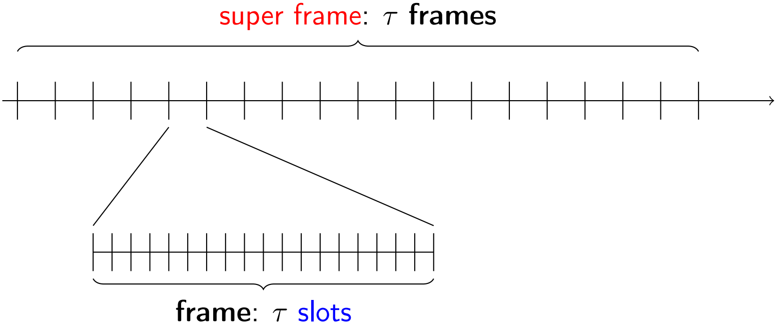
Med-Hoc-Net 2014
as brief announcement: SSS 2013

Thomas Petig, Elad M. Schiller, Philippas Tsigas

Outline

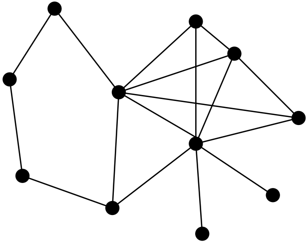
TDMA Frame

We divide the radio time is divided into: slots, **frames**, super frames:



The Problem

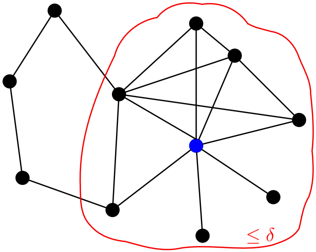
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

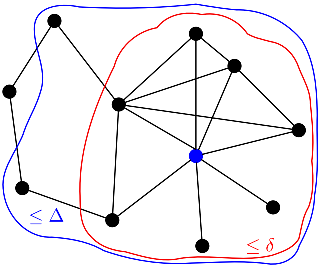
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

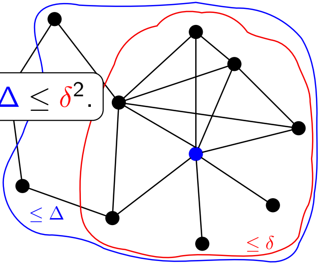
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

Given a communication graph:

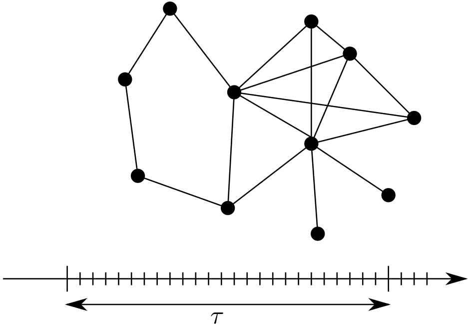


Note that: $\Delta \leq \delta^2$.

construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

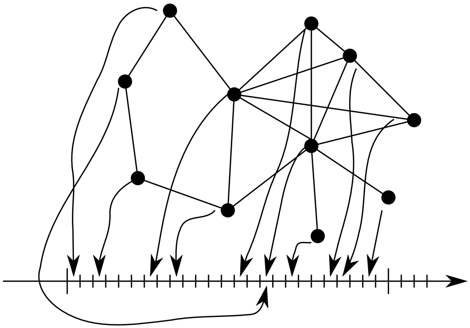
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

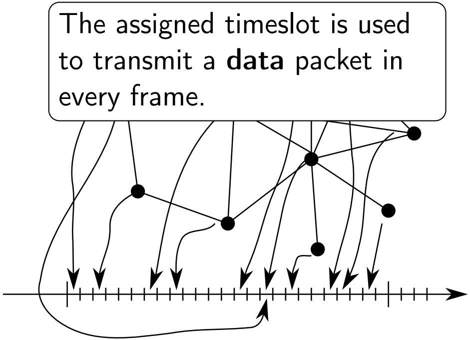
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

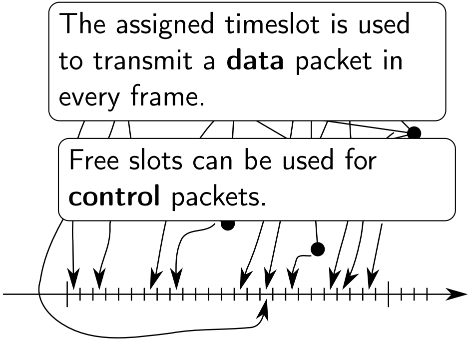
Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Problem

Given a communication graph:



construct a self-stabilizing, distance-2 coloring for TDMA slot allocation.

The Challenge

Collisions: concurrent transmissions might lead to packet omission.

We do not consider:

- external time references [Herman-Tixeuil ALGOSENSORS'04],
- external location references [Viqar-Welch ALGOSENSORS'09],
- collision detection,
- base stations for scheduling transmissions.

The Challenge

We have to show that communication is possible!

Collisions: concurrent transmissions might lead to packet omission.

We do not consider:

- external time references [Herman-Tixeuil ALGOSENSORS'04],
- external location references [Viqar-Welch ALGOSENSORS'09],
- collision detection,
- base stations for scheduling transmissions.

Our Approach

We focus on self-stabilizing algorithms that their converges considers both:

- clock synchronization, and
- time slot assignment.

Outline

Our Contribution

Basic limit on the bandwidth utilization of TDMA in wireless ad hoc networks:

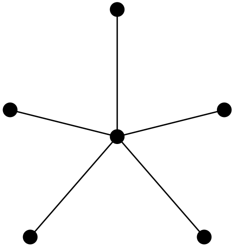
- $\tau < \max\{2\delta, \chi_2\}$, where χ_2 is the chromatic number for distance-2 vertex coloring.

Existent proves of collision-free self-stabilizing TDMA without assuming external reference availability.

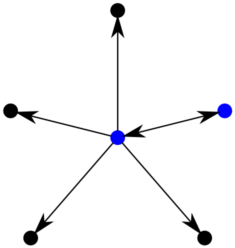
Lower Bound

Outline

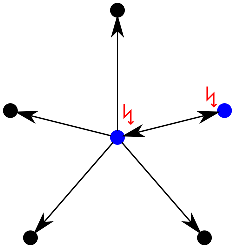
Lower bound



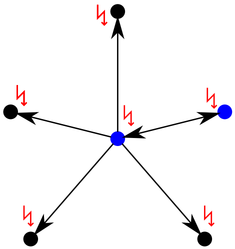
Lower bound



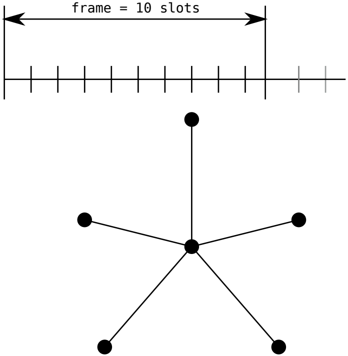
Lower bound



Lower bound

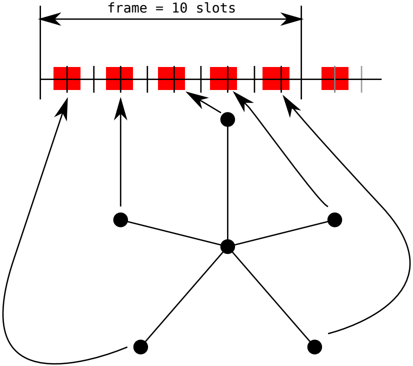


Lower bound



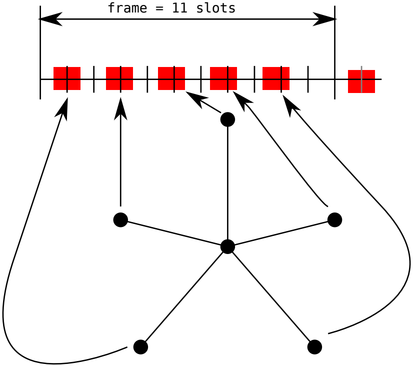
Lower bound

The δ leaves can block up to 2δ slots. This leads to 2δ as lower bound. Note that that there is a algorithm with frame size of $\mathcal{O}(2\Delta)$ [Busch, et al. Distributed Comp. '08].



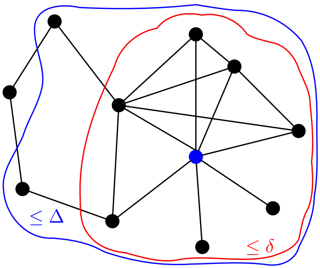
Lower bound

The δ leaves can block up to 2δ slots. This leads to 2δ as lower bound. Note that that there is a algorithm with frame size of $\mathcal{O}(2\Delta)$ [Busch, et al. Distributed Comp. '08].



Lower bound

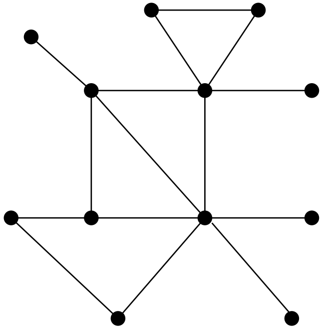
The δ leaves can block up to 2δ slots. This leads to 2δ as lower bound. Note that that there is a algorithm with frame size of $\mathcal{O}(2\Delta)$ [Busch, et al. Distributed Comp. '08].



Outline

Algorithm

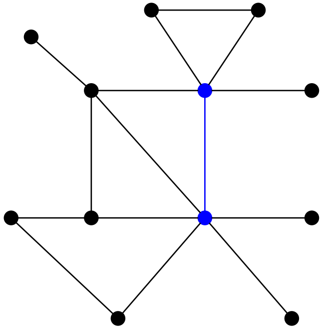
We focus on the communication to a single neighbor.



.

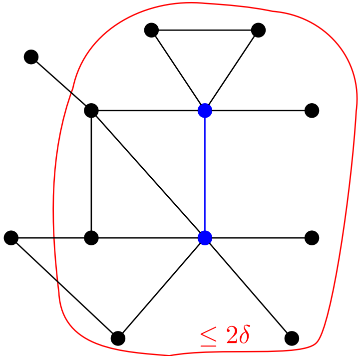
Algorithm

We focus on the communication to a single neighbor.



Algorithm

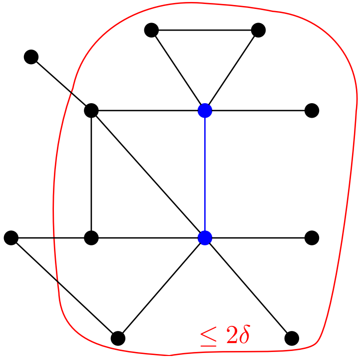
We focus on the communication to a single neighbor.



.

Algorithm

We focus on the communication to a single neighbor.



$\Rightarrow$ Communication is possible if $\tau \geq 4\delta$.

Algorithm

Upon packet reception:

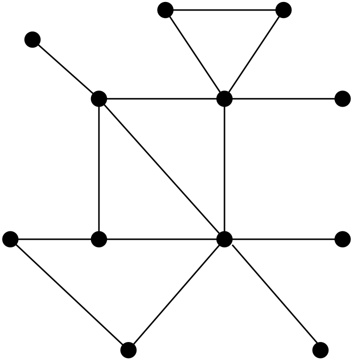
1. check clock (adjust and drop timeslot if higher)
2. check acknowledgement (drop timeslot if missing)
3. merge neighborhood

Upon timeslot:

1. If assigned TDMA timeslot then transmit.
2. If randomly chosen free timeslot
   2.1 transmit
   2.2 chose new random timeslot
   2.3 If no TDMA timeslot assigned then take this one

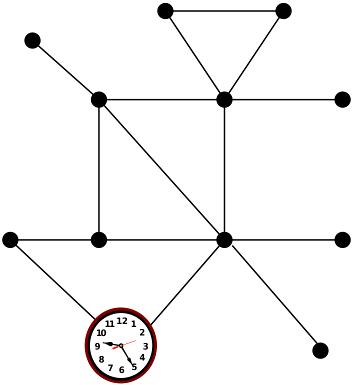
Algorithm

Clock synchronization.



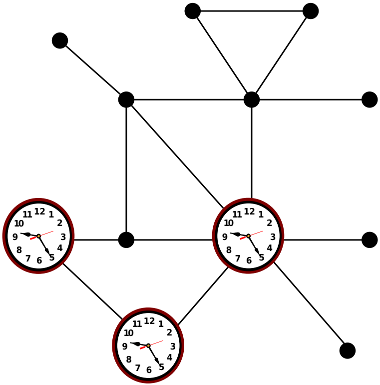
Algorithm

Clock synchronization.



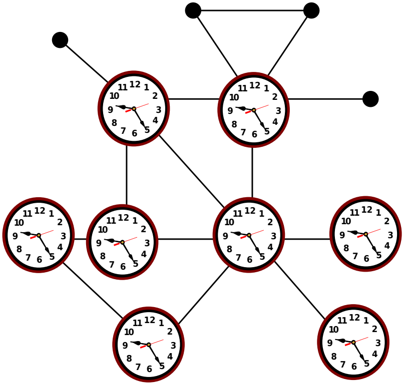
Algorithm

Clock synchronization.



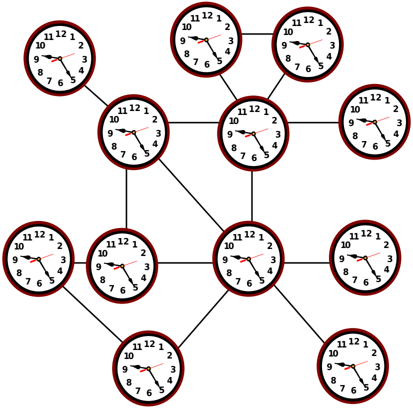
Algorithm

Clock synchronization.



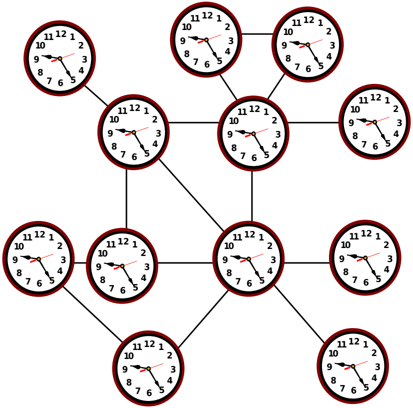
Algorithm

Clock synchronization.



Algorithm

Clock synchronization.



⇒ All clocks are synchronized!.

Algorithm

Upon packet reception:

1. ~~check clock (adjust and drop timeslot if higher)~~
2. check acknowledgement (drop timeslot if missing)
3. merge neighborhood

Upon timeslot:

1. If assigned TDMA timeslot then transmit.
2. If randomly chosen free timeslot
   2.1 transmit
   2.2 chose new random timeslot
   2.3 If no TDMA timeslot assigned then take this one

Algorithm

The convergence proof:

1. every node can reach a neighbor within an expected time,

See the technical report for more information.

Algorithm

The convergence proof:

1. every node can reach a neighbor within an expected time,
2. a converge-to-the-max approach for clock convergence [Herman and Zhang, SSS '08],

See the technical report for more information.

Algorithm

The convergence proof:

1. every node can reach a neighbor within an expected time,
2. a converge-to-the-max approach for clock convergence [Herman and Zhang, SSS '08],
3. each node gets a time slot that is unique within its neighborhood,

See the technical report for more information.

Algorithm

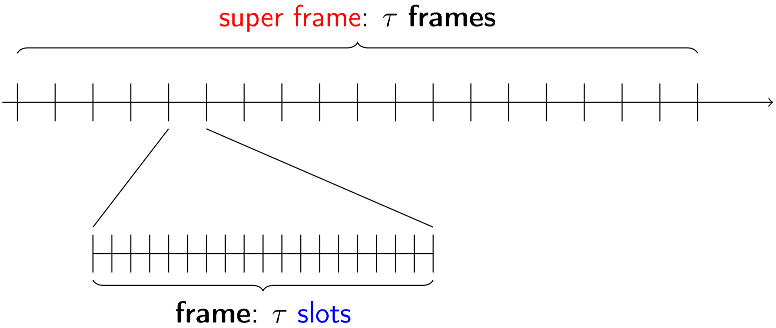
The convergence proof:

1. every node can reach a neighbor within an expected time,
2. a converge-to-the-max approach for clock convergence [Herman and Zhang, SSS '08],
3. each node gets a time slot that is unique within its neighborhood,
4. there are no packet collisions

See the technical report for more information.

TDMA Frame

We divide the radio time is divided into: slots, **frames**, super frames:



A slot can be used for a data packet or a control packet.
A data packet is send on a fixed slot within a **frame**.

Algorithm

During legal executions:

- TDMA time slots are aligned,
- each node successfully sends data packets once a frame,
- control packets do not collide.

Outline

Conclusions

Our system settings do not consider:

- external time reference,
- location reference,
- collision detection,
- base station.

Is it possible to combine the positive effects of TDMA and CSMA?
In our system settings:

- No, if the frame size is less than 2δ.
- Yes, if the frame size is larger than $\max\{4\delta, \chi_2\}$.

A preliminary implementation validates the setup.

Brief Announcement: Robust and Private Distributed Shared Atomic Memory in Message Passing Networks

PODC'15

Shlomi Dolev, Thomas Petig and Elad M. Schiller

Content

We focus on emulation **shared memory** in **message passing** networks.

Opportunity: Cadambe et al. (2014): A **coded shared atomic memory algorithm** for message passing architectures.

We are going to see how to provide

- **robustness** against semi-Byzantine attacks,
    - i.e., corruption of stored data,
- and **privacy** of the data.

in these networks.

Content

We focus on emulation **shared memory** in **message passing** networks.

Opportunity: Cadambe et al. (2014): A **coded shared atomic memory algorithm** for message passing architectures.

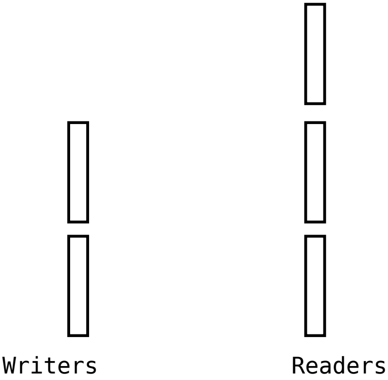
We are going to see how to provide

- **robustness** against semi-Byzantine attacks,
  - i.e., corruption of stored data,
- and **privacy** of the data.

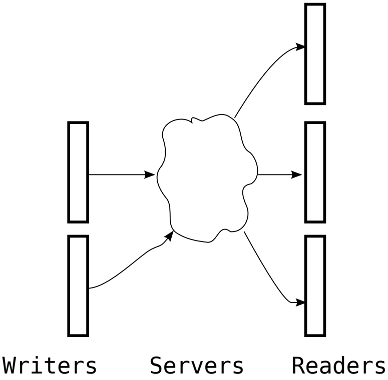
in these networks.

We use **Reed-Solomon codes**

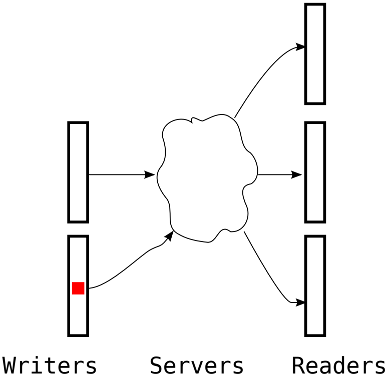
Multi Reader Multi Writer Shared Memory in Message Passing Networks



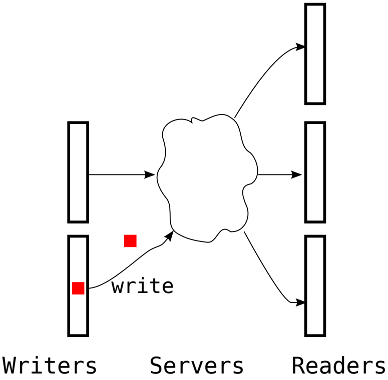
Multi Reader Multi Writer Shared Memory in Message Passing Networks



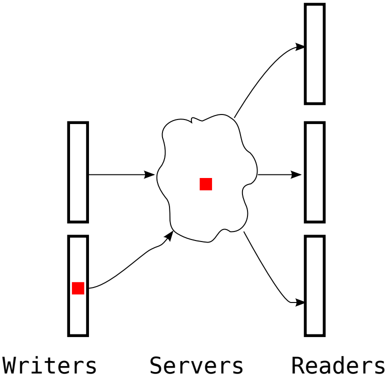
Multi Reader Multi Writer Shared Memory in Message Passing Networks



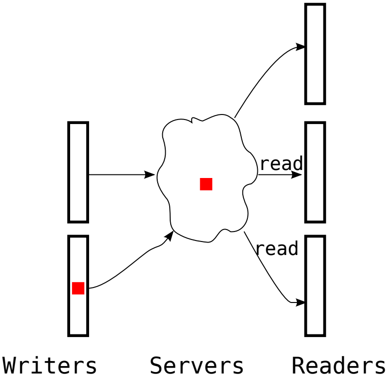
Multi Reader Multi Writer Shared Memory in Message Passing Networks



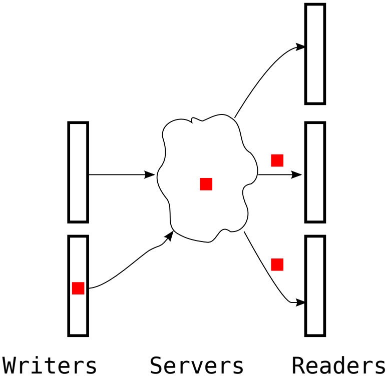
Multi Reader Multi Writer Shared Memory in Message Passing Networks



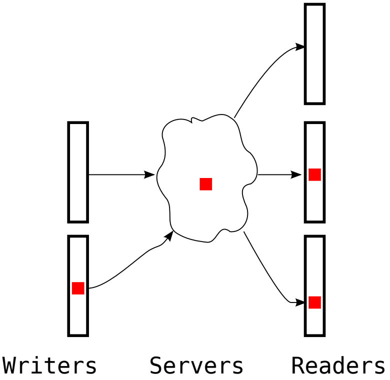
Multi Reader Multi Writer Shared Memory in Message Passing Networks



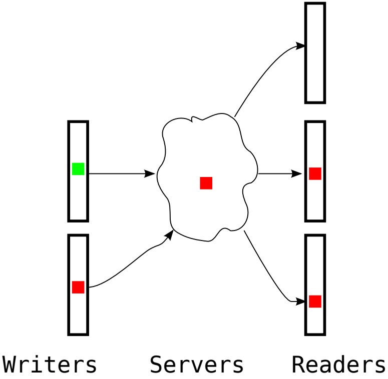
Multi Reader Multi Writer Shared Memory in Message Passing Networks



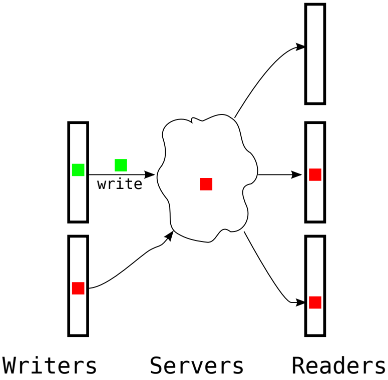
Multi Reader Multi Writer Shared Memory in Message Passing Networks



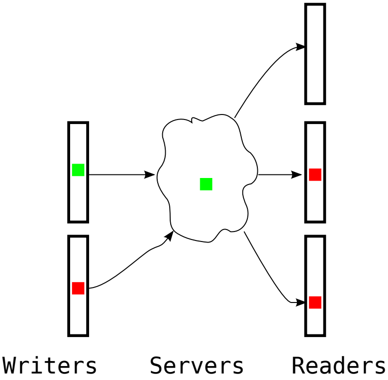
Multi Reader Multi Writer Shared Memory in Message Passing Networks



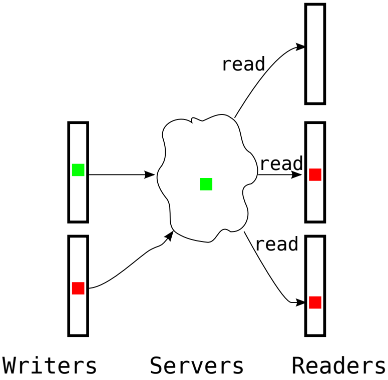
Multi Reader Multi Writer Shared Memory in Message Passing Networks



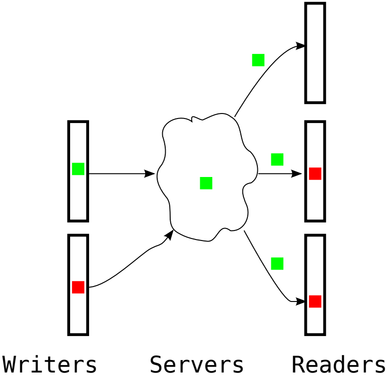
Multi Reader Multi Writer Shared Memory in Message Passing Networks



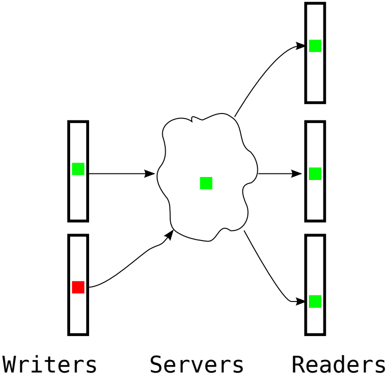
Multi Reader Multi Writer Shared Memory in Message Passing Networks



Multi Reader Multi Writer Shared Memory in Message Passing Networks



Multi Reader Multi Writer Shared Memory in Message Passing Networks



Multi Reader Multi Writer Shared Memory in Message Passing Networks

(Most) related work: Attiya, Bar-Noy, and Dolev (ABD), Cadambe et. al

Cadambe et al. address the following:

- atomicity and liveness and
- storage and communication costs.

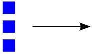
They solve atomicity and liveness in a ABD-like manner.

Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.

Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.

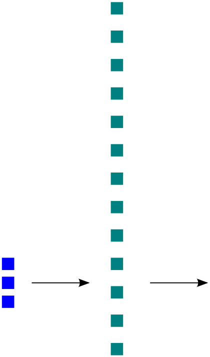


Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.

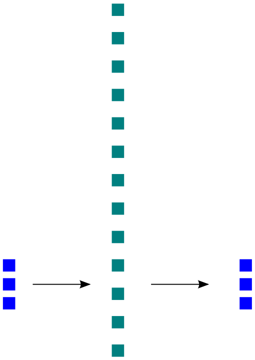
Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\to$ length N vector.
- tolerates $\leq N - k$ erasures.



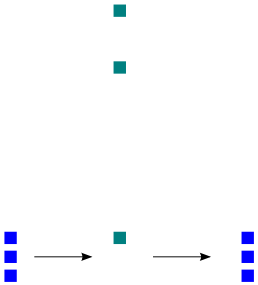
Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.



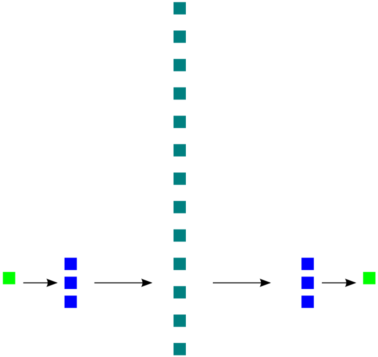
Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.



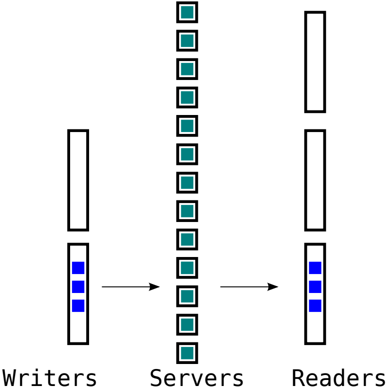
Erasure Coding: (N, k)-maximum distance separable codes

- length k vector $\rightarrow$ length N vector.
- tolerates $\leq N - k$ erasures.



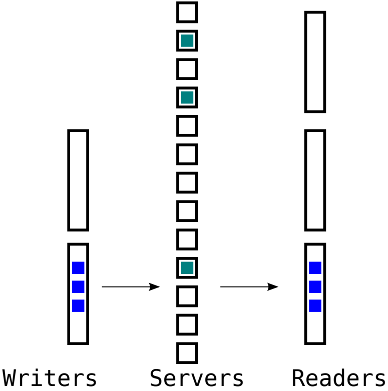
Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



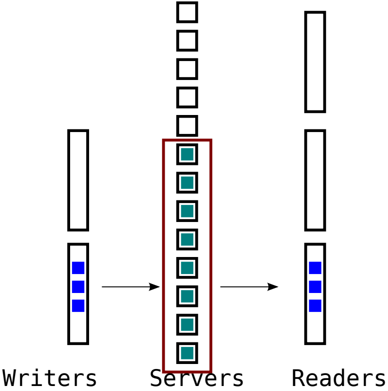
Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



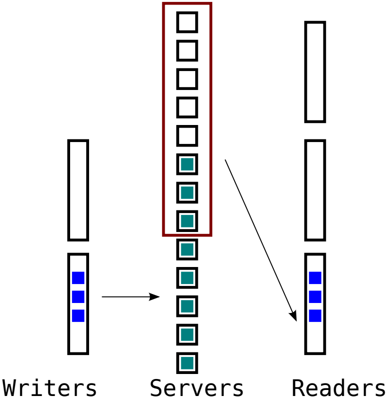
Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



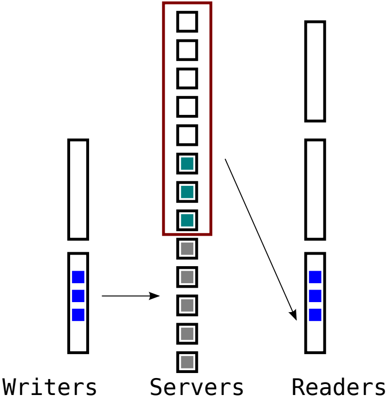
Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



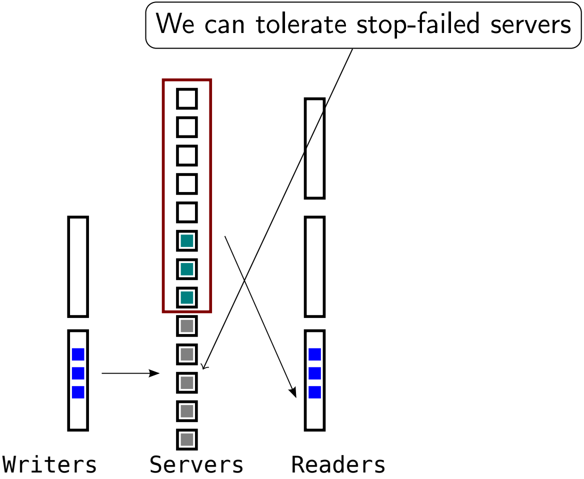
Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



Coded Atomic Storage Algorithm

- N servers.
- $\lceil \frac{N+k}{2} \rceil$-quorums.



Our contribution

We address:

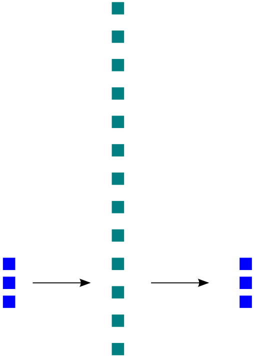
- Robustness against semi-Byzantine attacks.
- Privacy of the data.

We use

- (N, k)-**Reed-Solomon** codes and
- **Berlekamp-Welch** error correction.

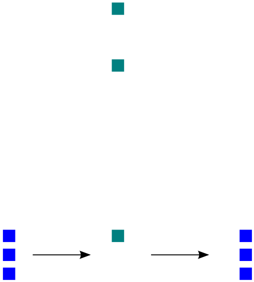
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



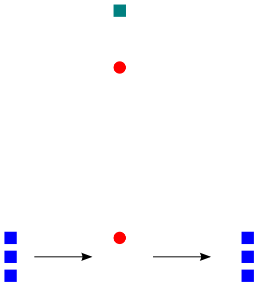
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



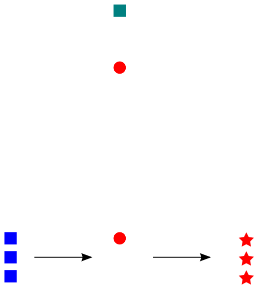
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



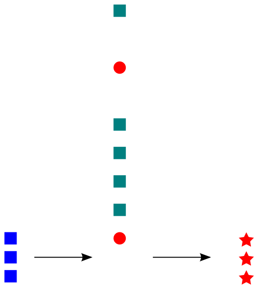
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



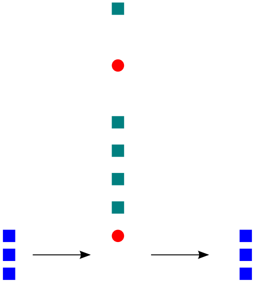
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



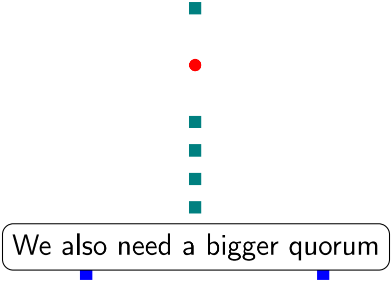
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



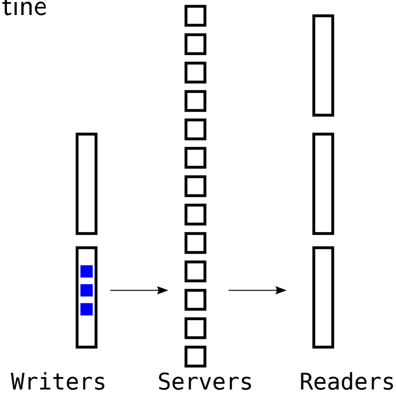
Robust and Private Coded Atomic Storage

- (N, k)-Reed-Solomon code.
- For e corrupt elements, we need to read $2e$ more elements.



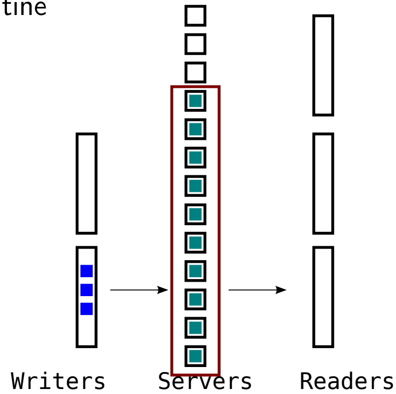
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



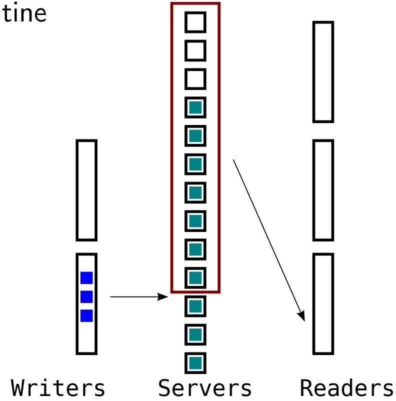
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



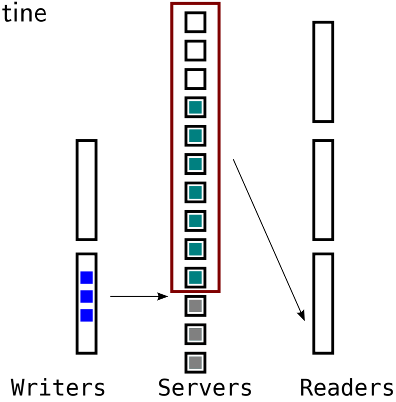
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



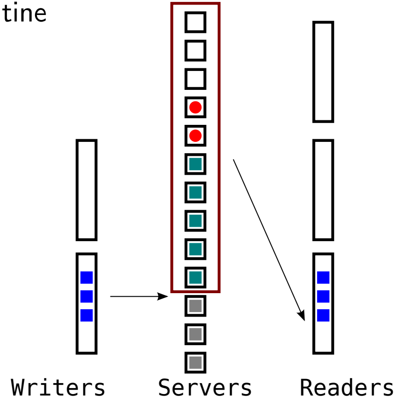
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



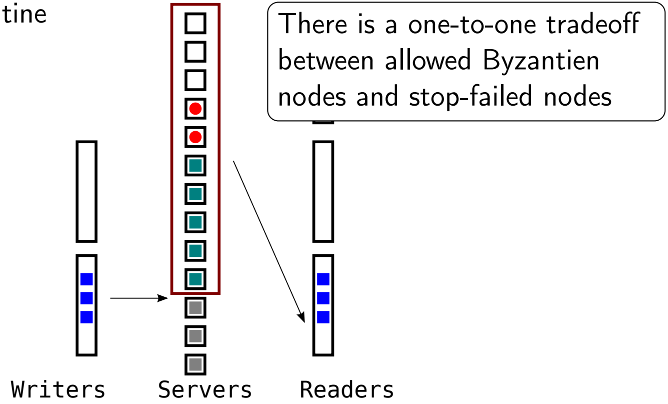
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



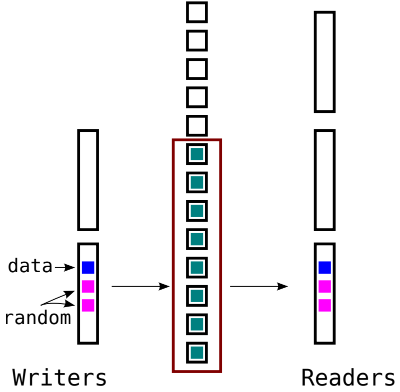
Robust and Private Coded Atomic Storage

- $\lceil \frac{N+k+2e}{2} \rceil$-quorums.
- Up to $f < N - \lceil \frac{N+k+2e}{2} \rceil$ failures.
- Up to e semi-Byzantine servers.



Robust and Private Coded Atomic Storage

- McEliece & Sarwate: Reed-Solomon codes are related to Shamir's **secret sharing**.
- Only sets of $\geq k$ server can reveal the secret.



Conclusion

Using special cases of coding (Reed-Solomon) and decoding (Berlekamp-Welch), we show:

- *robustness*, corrupted data by Byzantine server can be tolerated and
- *privacy*, even a small amount of server cannot restore the data.

Overall Conclusion

We have seen how to address some faults and failures.

- The proposed TDMA algorithm adds predictability and reliability to communication.
- The proposed coded atomic storage algorithm adds robustness and privacy to storge.