- Distributed Mutex

- Lamport Algorithm - about 3N msgs
  - improvements

- Mackawa Algorithm
- Responsibility

- a set of processes

- as stated in the

- process A for ...

- Request set process A as:

- Maclean
Mergemag → dict → empty list

<table>
<thead>
<tr>
<th>step set</th>
<th>= 0</th>
</tr>
</thead>
</table>

Counter

<table>
<thead>
<tr>
<th>step set</th>
<th>= 1</th>
</tr>
</thead>
</table>

dictionary

<table>
<thead>
<tr>
<th>step set</th>
<th>= N</th>
</tr>
</thead>
</table>

count

| step set | = N |
Nested set
intersection every other
except root set need
DME
In 2014nest

does he do that

< size < in

> from < in

< show that for

Use + from + preference please.
\[ \text{size} = (2nN - c) \]

\[ \text{degree} = 2 \]

\[ N \rightarrow \text{gen} \]

\[ \text{size} = \text{gen} \]
Discuss backed broke

Contribute big dataset

Let use mesh (edge fn

G ~ be G (edge fn

Label node by feedback
You got false. Decide global view X.
A \rightarrow \text{time delay} \rightarrow B
A → B

B ↓

↓

drop it

or forward

C

send probe

probe → msg
Distributed snapshots
5. Take a checkpoint of a set of processes on a distributed system.

Core dump/bugdump info (core)

Etc.
Construct algebraic expressions

Step 1

Step 2: Choose algebraic expressions

Step 3: Simplify

Very

Initial
Distributed Snapshot

Process Communication Model

\[ p_i \xrightarrow{\text{FIFO}} p_j \xrightarrow{\text{channel}} p_i \]

Pi sends message to Pi'
2 include main msg

3 proceed

User & Step Checkpoint

part of an communication

L spread msg not

make msg
Start node on all outgoing

Save states at C as T

Stop checks at T

If time + t arrives on

End process Pi for the

If successful msg arrives

Check C

Fix Pi
Do not broach marker.

Or after markers let us fit the edge of mass that came on the record, the edge of C on the

Not let time if needed arrive & if on channel,

if needed again &
A marker is transmitted once and only once.
The idea will not work.

No winners will remain.

The apple will decay from here.
Find visit
L vs. processed chessboard
L as chosen state
T \text{ or } \text{ Min} \text{ or } \text{ Max}
P \rightarrow A
\begin{cases}
\text{Send} (C_1, m) \\
\text{Recv} (C_2, m)
\end{cases}
\rightarrow A
\begin{cases}
\text{Send} (C_2, m) \\
\text{Recv} (C_1, m)
\end{cases}
\rightarrow A
\begin{cases}
\text{Send} (C_3, m) \\
\text{Recv} (C_4, m)
\end{cases}
\rightarrow A
\begin{cases}
\text{Send} (C_4, m) \\
\text{Recv} (C_3, m)
\end{cases}
\rightarrow A