THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN

AUTOMOTIVE
LINUX SUMMIT

Global Leader Election in Distributed Architecture

Dharmendra Kushwaha, NEC

THE LINUX FOUNDATION

Agenda

- Why Leader
- Why Leader Election
- Leader Election Mechanisms
- Requirements
- Challenges
- Solution

Why Leader

- An Organizer for some tasks
- Keeps nodes in synch.
- Responsible for controlling any changes in system.

Why Leader Election

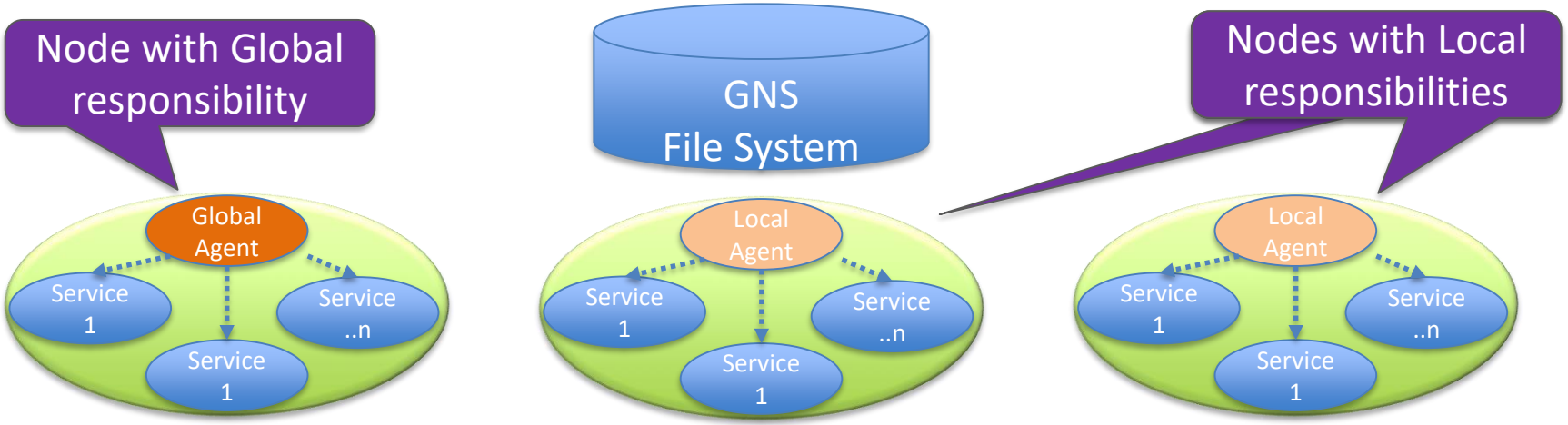
- Leader Election: A process of designating a single process as the organizer, coordinator, initiator or sequencer of some task distributed among several nodes or services.
  - The existence of a centralized controller greatly simplifies process synchronization
  - However, if the central controller breaks down, the service availability can be limited
  - The problem can be alleviated if a new controller (leader) can be chosen.

Leader Election Mechanism

- Bully Algorithm
- Ring Algorithm
- ..
- ..

Requirements

Requirements



Distributed service Infrastructure

Recovery and responsibility re-distribution.

Challenges

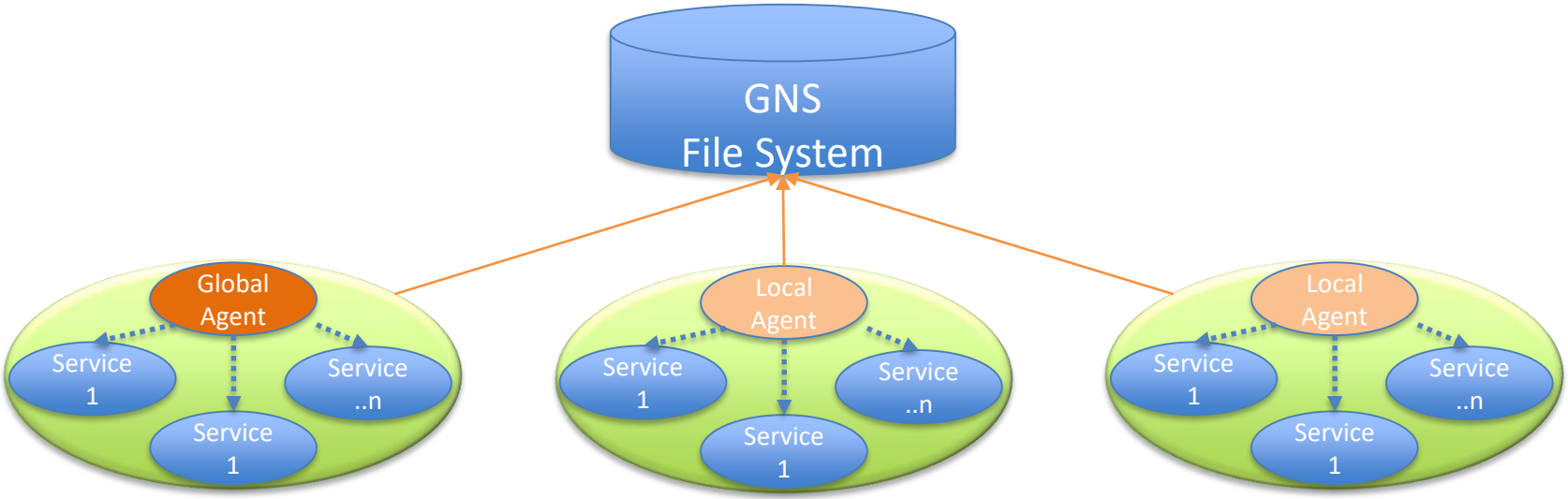
Time synchronization

- System timing may not be in sync
- Gaps can be more than a minutes.

Split Brain Problem

- May be more than one node can start behaving like leader.
- May be no leader.
- Most of the solutions usage third party components.

GFS accessibility is must

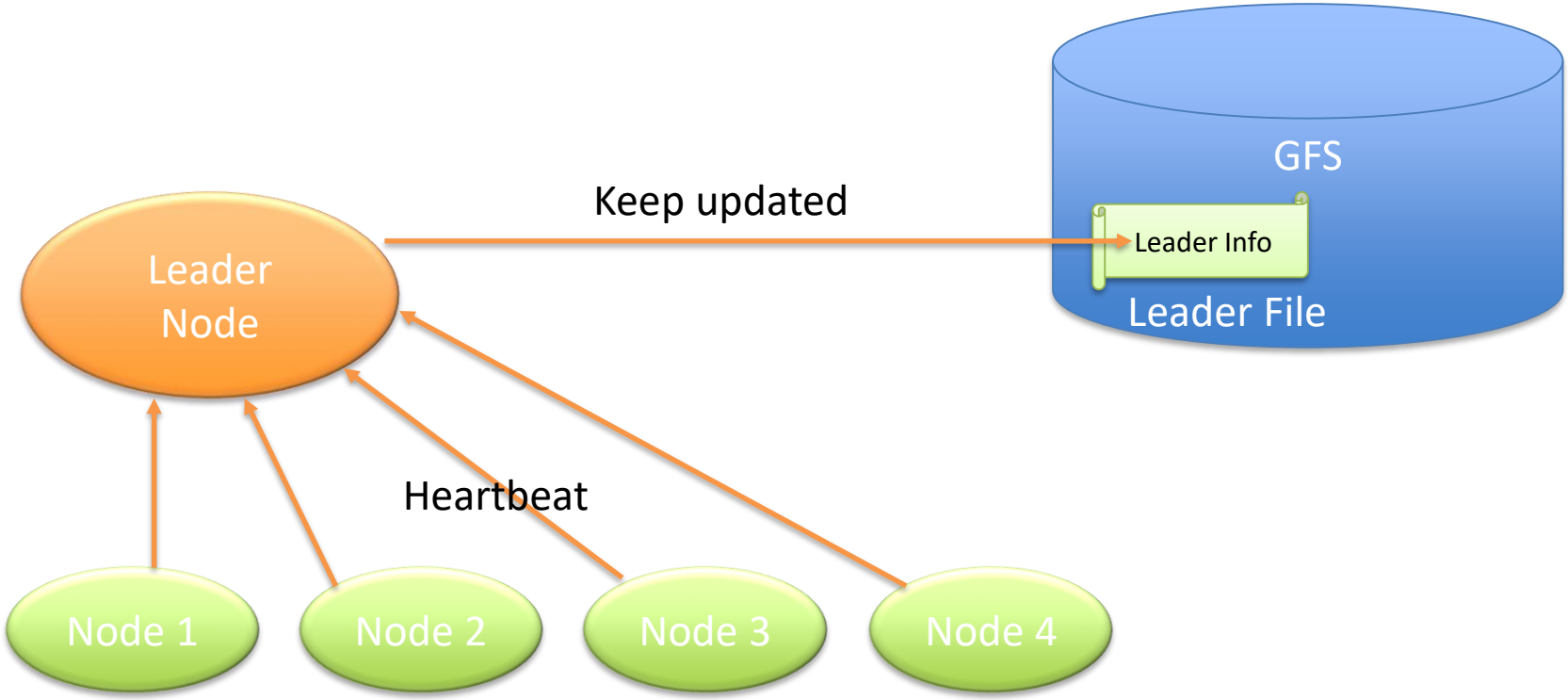


Solution

How encounter those challenges

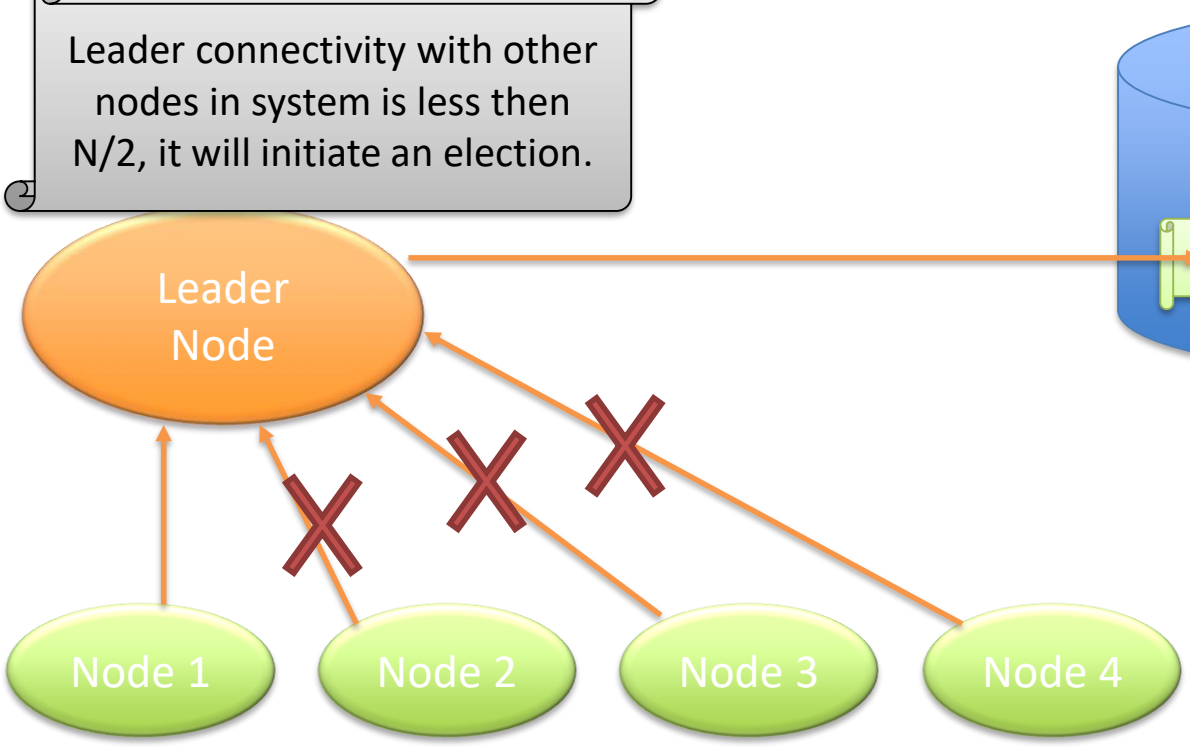
- Leader must have both network & filesystem connectivity.
- Election coordinator for time sync.
- GNS FileSystem(GFS) to handle split brain.

Leader Prerequisite



Election Trigger Points

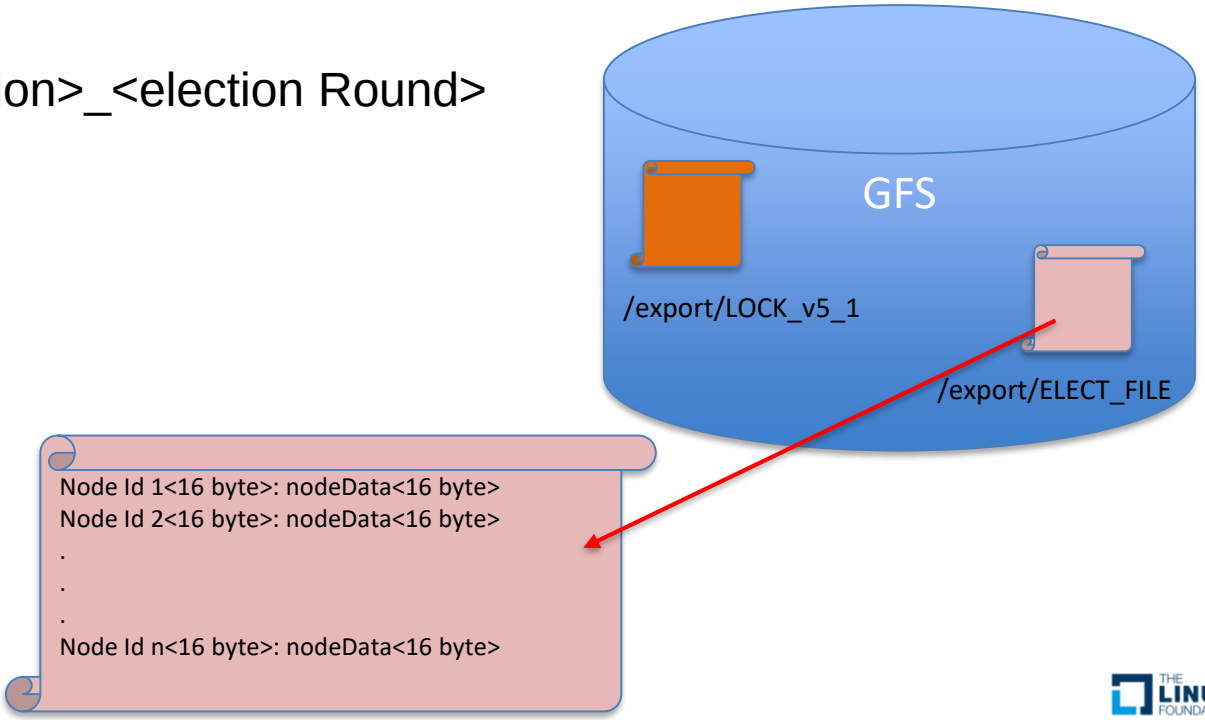
Leader connectivity with other nodes in system is less then N/2, it will initiate an election.



Node-x is not able to send heartbeat to global leader, it will check the last update time of Leader info file. If file is not updating, node will assume that leader is down.

Election configs

- Lock File:
  - LOCK_<GL version>_<election Round>
- Election File:
  - ELECT_FILE



Election configs..

- Timeouts:
  - T_{el}: Election Timeout
  - T_{p_el}: Participant nodes timeout
  - $T_{e\_file\_wait}$: Election file write timeout
  - T_{notify}: Notification time out

Election Algorithm

GL_Election:

1. Get Election version.
2. Check for Lock file
    a. If lock file exist:
        i. Is file's Leader version is older.
            1. Remove existing lock file.
            2. Create new lock file.
                a. If success, proceed as election manager. Otherwise, proceed as participant.
        ii. Is Leader version is same.
            1. Is file's election round is older.
                a. Update Lock file.
                    i. If success, proceed as election manager. Otherwise, proceed as participant.
            2. Otherwise, proceed as participant.
        iii. Otherwise, proceed as participant.

Election Algorithm..

Proceed as Election Manager:

1. Calculate file size.
2. If election participant file exist on GFS.
    a. Remove existing file.
3. Create a temp file and fill it with null bytes by its size.
4. Move temp to Election participant File
5. Write own Id and data(i.e. connection count) on specific offset in file.
6. Start Timer T_{el}, and wait for this timeout.
7. Disable write After time T_{el} (move election participant file to temp elect file)
8. Elect Leader on the basis of data(i.e max connectivity & higher id).
9. Notify newly elected leader to take the ownership, and Start timer T_{notify}.
10. If got response from leader within T_{notify} timeout or leader info file is updated.
    a. Leader election done, Clean lock & election participant file.

Otherwise call GL_Election.

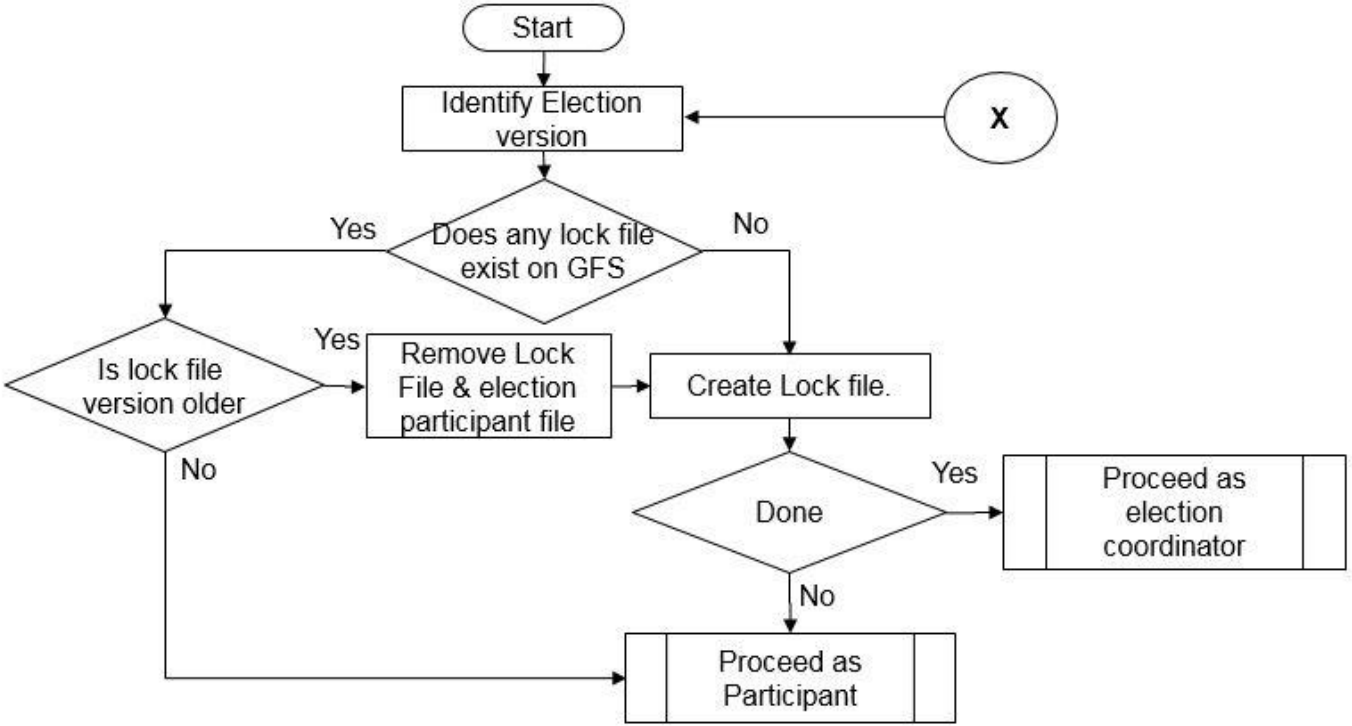
Election Algorithm..

Proceed as Election Participant:

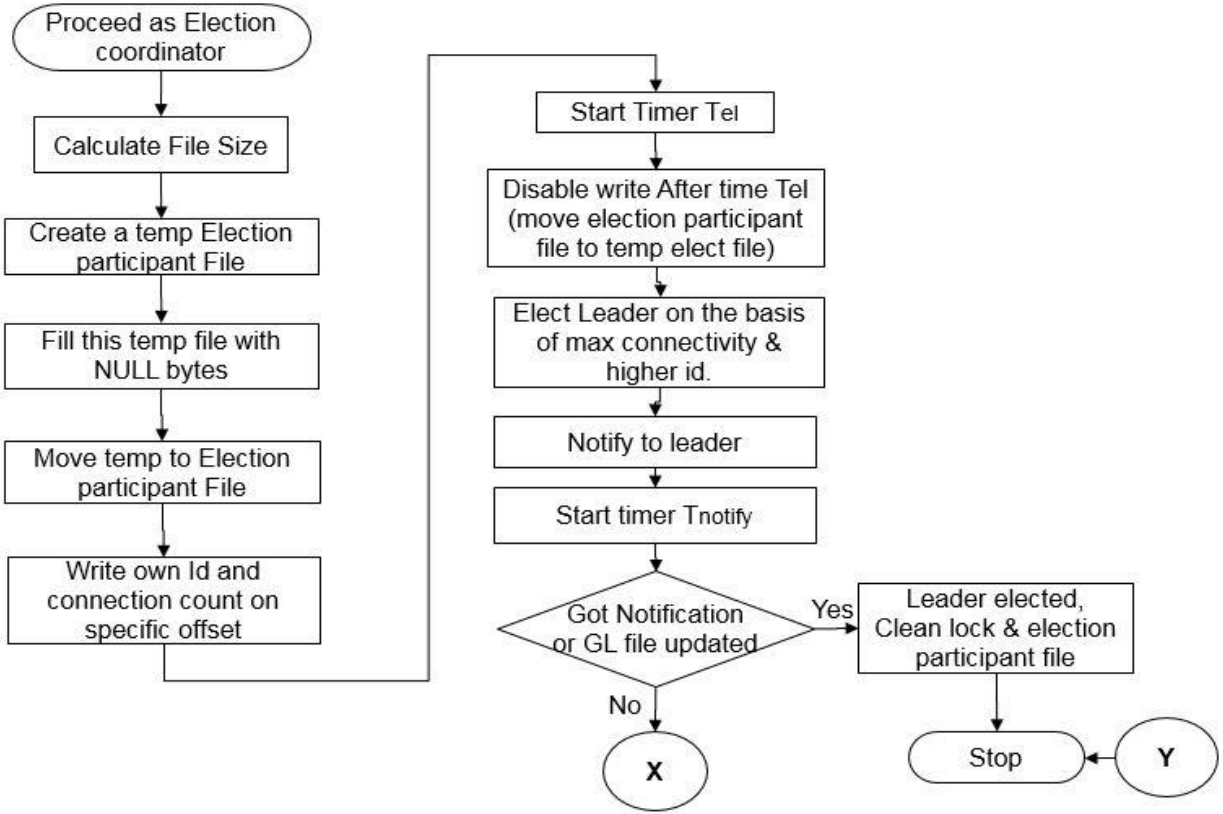
1. If no Election participant file exist on GFS.
    a. If Temp Election participant file exist.
        i. Election window closes, goes out of election.
    b. Otherwise, Wait for $T_{e\_file\_wait}$ time.
    c. After $T_{e\_file\_wait}$ time if Election Participant file is still not created.
        i. Call GL_Election.
2. Write own id & connectivity count in election participant file
3. Start Timer Tp_el and wait.
4. If during timer Tp_el got Leader notification
    a. Stop timer.
    b. If Leader file updated.
        i. Leader election done.
        ii. Update own status and exit.
    c. Otherwise:
        i. Take Leader ownership (i.e. update Leader file)

Respond to election manager and exit.

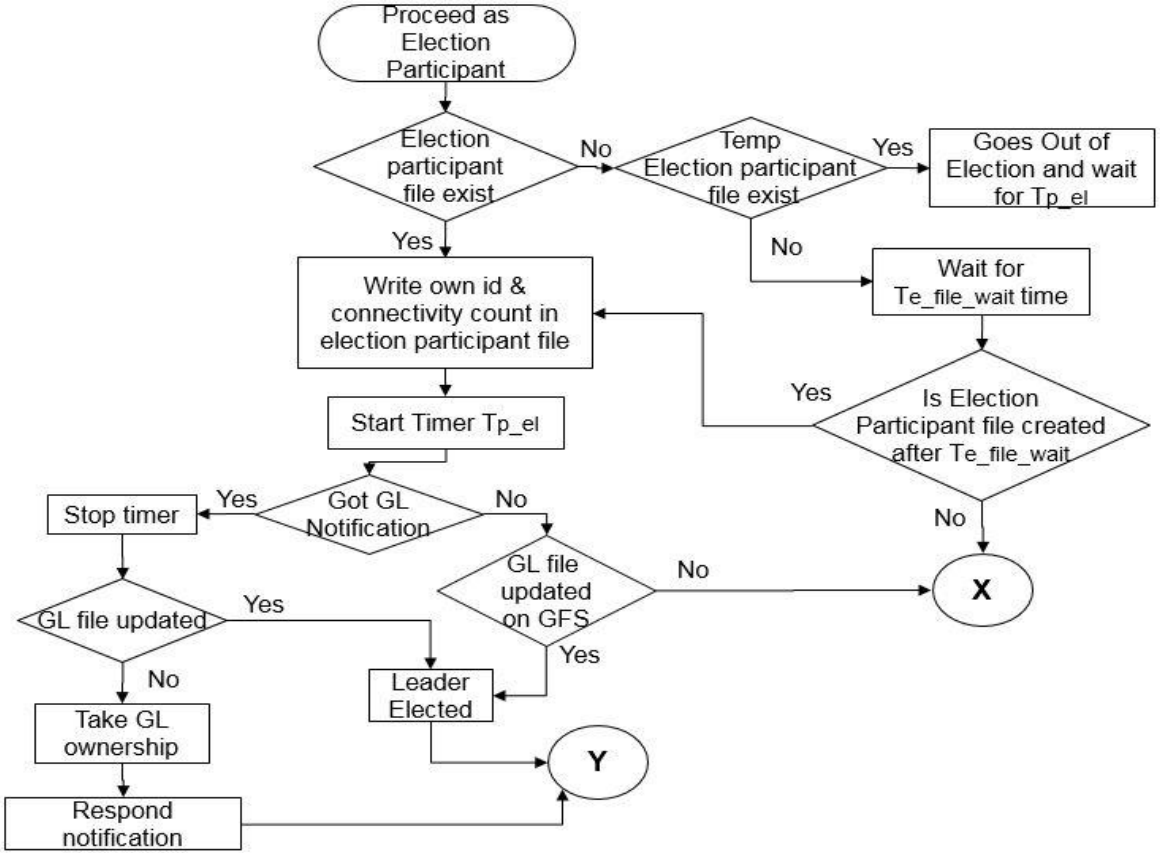
Election Algorithm: Flow Chart



Election Algorithm: Flow Chart..



Election Algorithm: Flow Chart..



Thank You

Q&A?

Dharmendra.Kushwaha@gmail.com

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT JAPAN

AUTOMOTIVE LINUX SUMMIT

THE LINUX FOUNDATION