Applications today tackle several areas of interest, such as:
• Surveillance of premises or of power equipment installations (towers and power transmission lines);
• Meter reading and consumption monitoring, residential and commercial;
• Power demand and supply management, including integrated renewable energy sources; and
• Maintenance of power supply systems, by detecting line faults and failures.

The IoT-aided transmission tower protection system contains various sensors which generate early warnings of threats to high voltage transmission towers, enabling quick responses. The sensors include vibration sensors, anti-theft bolts, a leaning sensor and a video camera. These sensors and the sink node form a WSN. The sensors detect any threat, and send the relevant signals to the sink node. The sink node receives these signals from the sensors, processes them into data and transmits the data to the monitoring center through the Internet or any other public/private communication network.
Sensors measuring conductor galloping, wind vibration, conductor temperature, micro-meterology and icing can now be used to achieve real-time online monitoring of power transmission lines.

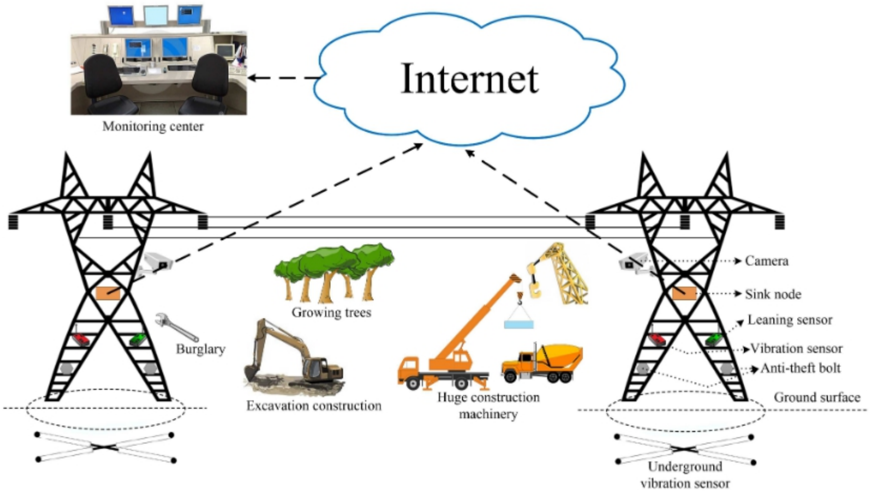


Fig. 4. IoT-aided transmission tower protection system for the safety of transmission towers from the threats of buglary, natural disasters, barbaric construction and growing trees. This system is comprised of a sink node and various sensors which generate early warnings to the monitoring centers about the threats to high voltage transmission towers. The sensors include two vibration sensors (one underground in the base of tower and other on the tower), anti-theft bolts, leaning sensor and video cameras [118].

The grid uses sensors for collecting data about temperature, humidity, noise etc. which ensures monitoring and secure operation of distribution grid. Together, each the sensors in this grid shall form groups of clusters of actors that interact with each other. In order for these clusters to work properly, fault-tolerance, vertical and horizontal scalability and failure recovery should be taken in account.

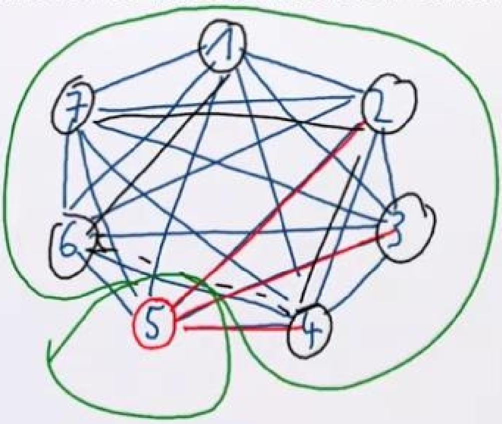
Cluster needs Failure Detection

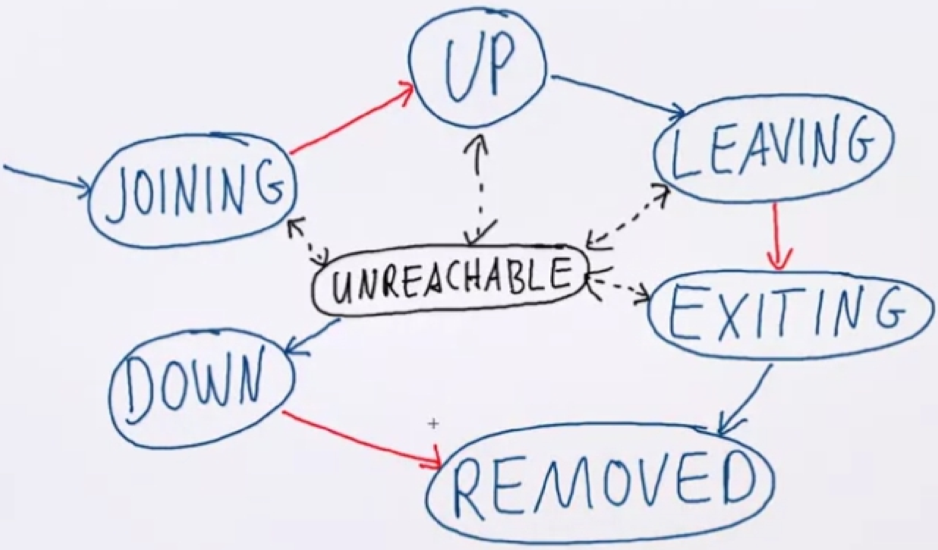
Consensus is unattainable if some members are unreachable.

Every node is monitored using heartbeats from several others.

A node unreachable from one other is considered unreachable for all.

Nodes can be removed to restore the cluster consensus.





Applying it to ClusterWorker

```
val cluster = Cluster(context.system)
cluster.subscribe(self, classOf[ClusterEvent.MemberUp])
val main = cluster.selfAddress.copy(port = Some(2552))
cluster.join(main)

def receive = {
  case ClusterEvent.MemberUp(member) =>
    if (member.address == main) {
      val path = RootActorPath(main) / "user" / "app" / "receptionist"
      context.actorSelection(path) ! Identify("42")
  case ActorIdentity("42", None)       => context.stop(self)
  case ActorIdentity("42", Some(ref)) => context.watch(ref)
  case Terminated(_)                   => context.stop(self)
}
```

The Ask Pattern

```
import akka.pattern.ask

class PostsByEmail(userService: ActorRef) extends Actor {
  implicit val timeout = Timeout(3.seconds)
  def receive = {
    case Get(email) =>
      (userService ? FindByEmail(email)).mapTo[UserInfo]
        .map(info => Result(info.posts.filter(_.email == email)))
        .recover { case ex => Failure(ex) }
        .pipeTo(sender)
  }
}
```

Result Aggregation

```
class PostSummary(...) extends Actor {
  implicit val timeout = Timeout(500.millis)
  def receive = {
    case Get(postId, user, password) =>
      val response = for {
        status <- (publisher ? GetStatus(postId)).mapTo[PostStatus]
        text   <- (postStore ? Get(postId)).mapTo[Post]
        auth   <- (authService ? Login(user, password)).mapTo[AuthStatus]
      } yield
        if (auth.successful) Result(status, text)
        else Failure(”not authorized”)
      response pipeTo sender
  }
}
```

Summary

Asynchronous message passing enables vertical scalability.

Location transparency enables horizontal scalability.