

Internet of Things



MATERI 8: IoT Robustness and Reliability

What we learn today ...

- The IoT Robustness and Reliability
- IoT Characteristics and Reliability Issues
 - Failure Scenarios
 - Reliability Challenges
 - Privacy and Reliability
 - Interoperability Device
- Addressing Reliability
 - Nullifying Impact Of Fault
 - Error Detection
 - Fault Prevention

reliability

/re·li·abil·i·ty/ noun

Being able to be trusted to do what is expected or has been promised.

Introduction

Building a reliable computing system has always been an important requirement for the business and the scientific community.

Two Important requirements :

1. Reliability : how long a system can operate without any failure
2. Availability : percentage how long a system can operate without any failure

Two Important variable:

1. Mean time to failure (MTTF)
2. Mean time to Recover (MTTR)

Infrastructure Fault

1. The devices can fail due to the physical condition and interference with the environment
2. Unexpected inputs from the external environment
3. Keeping a small form-factor in mind rather than making them highly fault-tolerant
4. Limited compute time and unexpected termination of a computation

Interaction Fault

1. The network fail
2. Impedance mismatch caused by the heterogeneity of the devices
3. Unexpected workload

Fault in Service Platform

1. Many transient faults can be due to off-the-shelf components
2. Provide incorrect data

The challenges are :

1. Making Service Available to User
2. Serviceability of IoT System
3. Reliability at Network Level
4. Device Level Reliability

The two important :

1. Identity management
2. Proving identity on-demand

Interoperability device has one and most important issue Lack of standardization



Approach for reliable system is to take fault prevention or fault tolerance to ensure reliability and availability

Fault-tolerance is achieved :

1. Nullifying the impact of an error
2. Error detection
3. Fault Prevention

Redundancy in Service Platform Design

1. Load balancing across multiple databases
2. Data replication

Redundancy in M2M Topology

1. Storing replicas of the local data and computation on the neighboring devices
2. Monitors another device(s) through a heartbeat mechanism

Graceful Degradation

1. Software Design
2. Performability Model

Four techniques

1. Watchdog
2. Heartbeat
3. Exception Handling
4. Recovery Through Restart

- This is a lightweight timer which runs separately from the main process.
- If the main process does not periodically reset the timer before it expires, the process is assumed to have a control flow error (the correct flow of control would have reset the timer).
- In such a case, a hardware-implemented timer can generate an interrupt that can trigger a recovery routine. The recovery procedure can :
 - ✓ Restart the process or the system from its checkpointed state
 - ✓ Invoke an appropriate recovery routine

- In this approach, a node in the network sends a message with a payload (ie, meaningful results, execution progress) to indicate that it is alive.
- If the heartbeat signal is not received within the prescribed time, the monitoring component assumes that the application running on the node has failed.
- Many modern distributed systems use adaptive heartbeat mechanism where the heartbeat monitoring component estimates the heartbeat message roundtrip time (RTT).

- The basic premise behind an error detection technique is that an application satisfies a set of properties for all correct executions of the application.
- If the property is not satisfied for any run of the program due to a fault, exception occurs in the program.

- At a minimum, autonomous devices as well as network devices need to have a basic restart mechanism, which can be triggered by the watchdog timer, or by the service platform.
- Restart is a useful technique for autonomous devices to recover from any transient error.
- However, it is often costly to have the entire system restart. Recently, a technique known as micro-reboot has been proposed where a module can be selectively restarted rather than the entire system.

Four techniques

1. Failure Prediction
2. Improving Communication Reliability
3. Failure Prevention by Service Platform
4. Improving Energy Efficiency

- A generic approach for any failure prediction is to define a set of invariants the application must satisfy when it is in operation.
- If these invariants are broken, the application can encounter an imminent failure.
- If these invariants are modeled properly, it is possible to implement a monitor that can watch for a failure of any invariant and can take a preventive action.

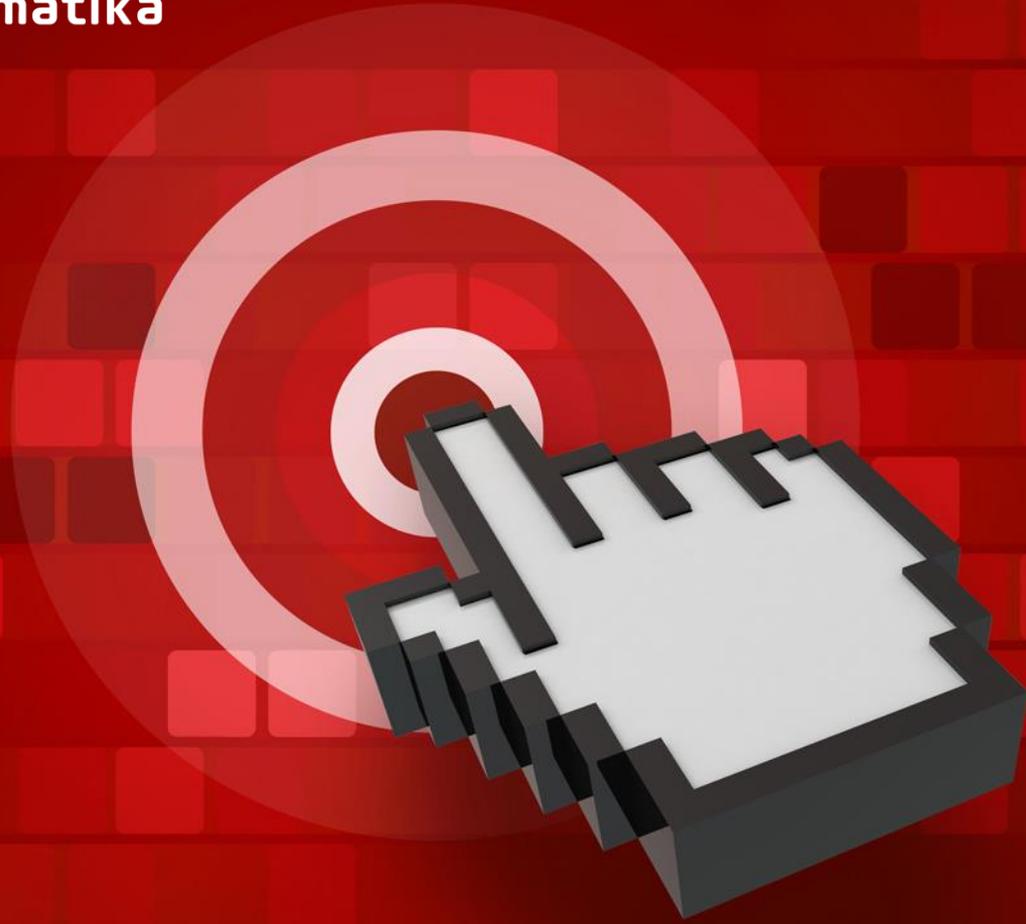
- In many IoT application scenarios, the communication of machines and sensors without human intervention is a key requirement.
- To achieve this goal without failure, the reliability of the communication system that connects the sensors together plays a key role in the overall system reliability.
- A communication protocol provides the needed redundancy in IoT, it is also important to optimize the number of hops, without sacrificing the reliability so that the energy consumption is under control.

- A sensor topology can introduce redundancy to reduce the single point of failure, the sensors may not have the overall sense of the topology to improve their longevity or to modify the next course of action based on the collective information generated out of the topology.
- The service platform, equipped with the knowledge of the overall topology, can take a better decision to prevent any failure.

- Since an IoT-based system is supposed to be operational for a long time, it is important that the autonomous devices can run on a battery power for a long time.
- To achieve this, a holistic approach to optimize the energy consumption needs to be considered to prevent any failure due to the unavailability of the battery power.
- In the context of IoT, the data quality management has become all the more relevant specifically when the quality of data collected by the sensors has a direct impact on the energy consumption.



Fakultas Informatika
School of Computing
Telkom University



THANK YOU