

Introduction

CS60002: Distributed Systems



Pallab Dasgupta

Dept. of Computer Sc. & Engg.,

Indian Institute of Technology Kharagpur

Books

- **Advanced Concepts in Operating Systems**
 - **Mukesh Singhal and Niranjan G. Shivaratri**
McGraw Hill International Edition

- **Introduction to Distributed Algorithms**
 - **Gerard Tel**
Cambridge University Press
Available in the CSE Dept Library (Acc No: I-455)

What is a distributed system?

A very broad definition:

- *A set of autonomous processes communicating among themselves to perform a task*

Issues:

- Un-reliability of communication
- Lack of global knowledge
- Lack of synchronization and causal ordering
- Concurrency control
- Failure and recovery

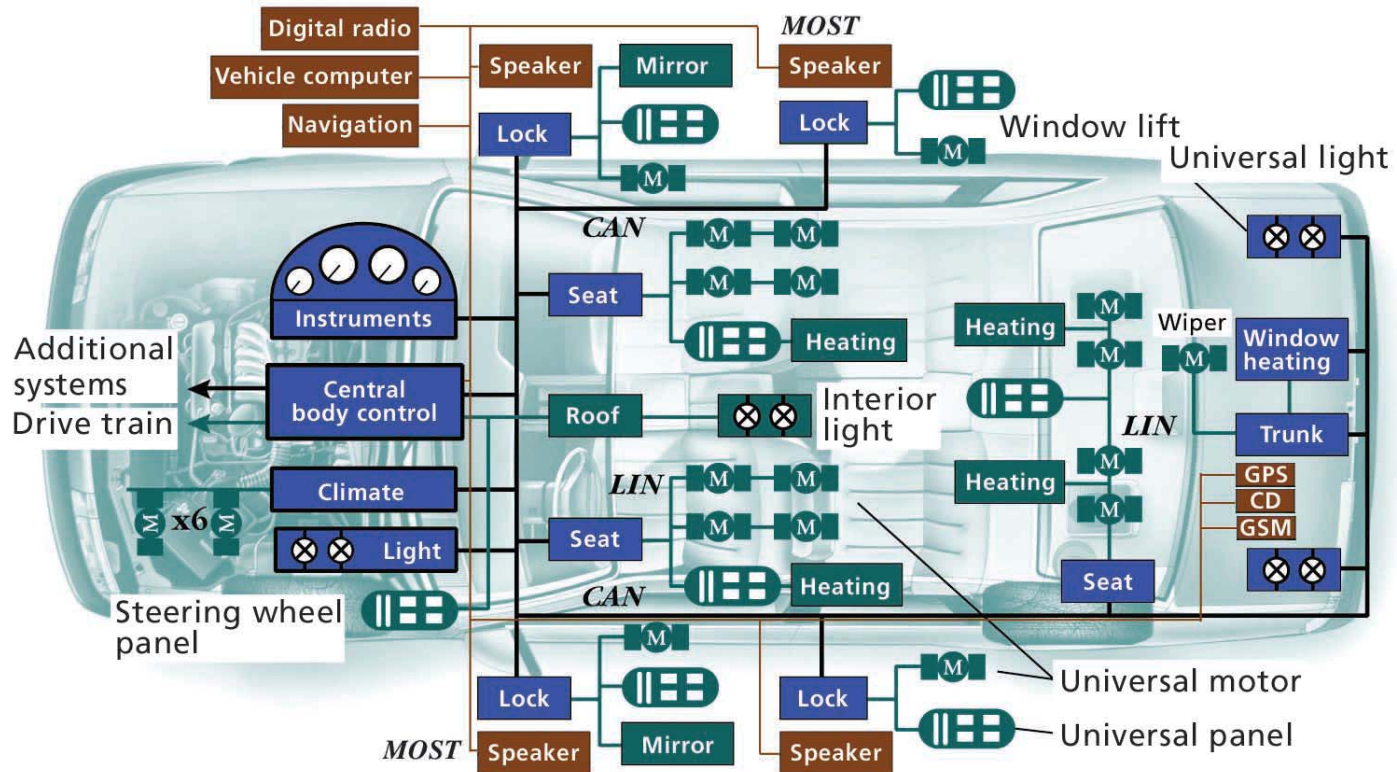
Advantages

- **Resource Sharing**
- **Higher Performance**
- **Fault Tolerance**
- **Scalability**

Examples of problems

- **Reliable communication – *Theoretically impossible?***
- **Muddy forehead and related problems**
- **Concurrency problems**

Example: Automotive Control



- CAN Controller area network
- GPS Global Positioning System
- GSM Global System for Mobile Communications
- LIN Local interconnect network
- MOST Media-oriented systems transport

Source: Leen and Hefferman, IEEE Computer, Jan 2002

Why is it hard to design them?

- The usual problem of concurrent systems:
 - Arbitrary interleaving of actions makes the system hard to verify

+

- No globally shared memory (therefore hard to collect global state)
- No global clock
- Unpredictable communication delays

Models for Distributed Algorithms

- **Topology**: Completely connected, Ring, Tree etc.
- **Communication**: Shared memory / Message passing
(reliable? Delay? FIFO/Causal? Broadcast/multicast?)
- **Synchronous/asynchronous**
- **Failure models**: Fail stop, Crash, Omission, Byzantine...
- ***An algorithm needs to specify the model on which it is supposed to work***

Complexity Measures

- **Message complexity: no. of messages**
- **Communication complexity / Bit Complexity: no. of bits**
- **Time complexity:**
 - **For synchronous systems, no. of rounds**
 - **For asynchronous systems, different definitions are there.**

Some Fundamental Problems

- **Ordering events in the absence of a global clock**
- **Capturing the global state**
- **Mutual exclusion**
- **Leader election**
- **Clock synchronization**
- **Termination detection**
- **Constructing spanning trees**
- **Agreement protocols**