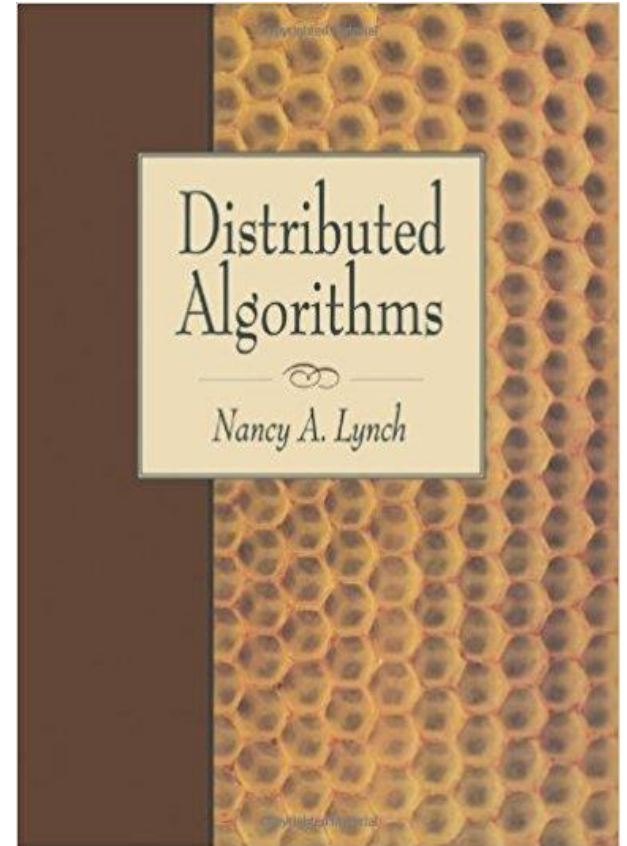


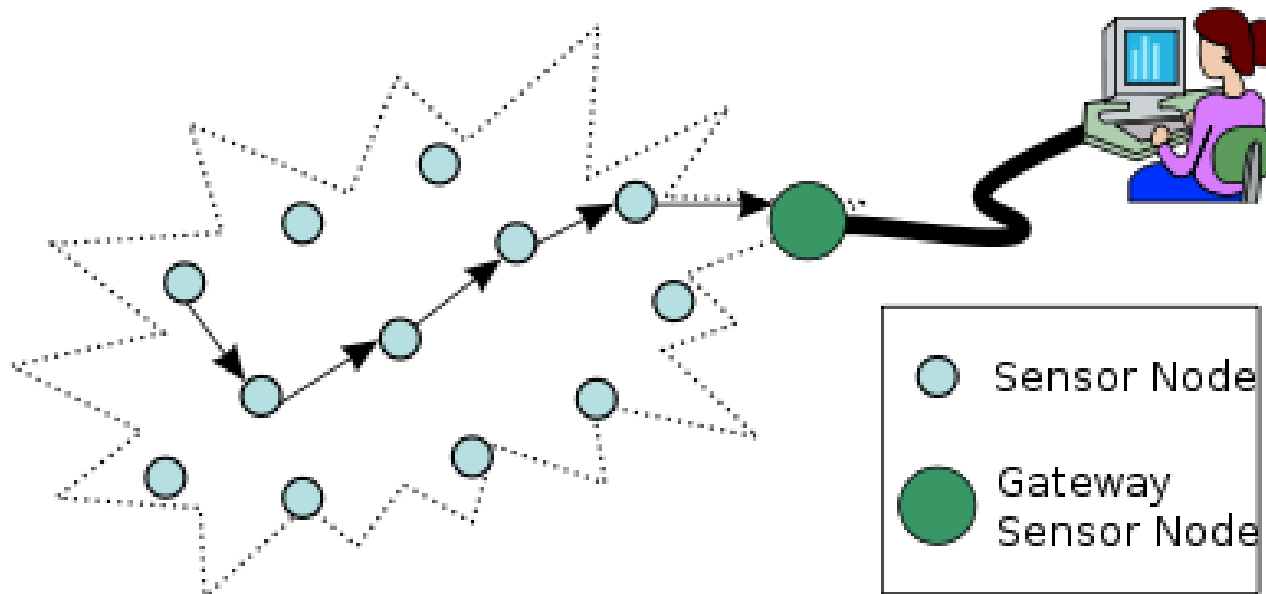
References

Nancy A. Lynch, **Distributed Algorithms**.
Morgan Kaufmann Publishers Inc., First
edition.



What is a distributed system?

It is a **network** of **computing elements** interacting with one another to achieve a goal.



(The **nodes** are processes, and the **edges** are communication channels.)

A channel may be physical (wired, wireless) or logical

Examples

- The World Wide Web runs on the Internet and provides service
- eBay for internet-based auction
- Sensor networks for monitoring environmental data
- BitTorrent (P2P network) for downloading video / audio
- Skype, FaceTime for making audio and video communication
- Facebook, Twitter, Google (the oxygen of many people)
- Process control networks in engineering factories
- Network of mobile robots collectively doing a job
- Distance education, net-meeting etc.
- Netbanking
- Applications on Vehicular networks (VANET)

More Examples

- Domain Name System (DNS)
 - Distributed lookup table of hostname to IP address
- Email servers (SMTP)
- Phone networks: Land line and cellular
- Cars network electronic components via CANbus
- Traffic light controllers
- Train control networks
- Airplanes
 - Avionics use RS232 serial links
 - Air traffic control uses verbal communication

What is Distributed System?

There is no universally accepted definition of distributed system.

Distributed Systems

- From “Distributed algorithms”
 - N. A. Lynch
 - Morgan Kaufman, ISBN 1-55860-348-4
- “*Distributed algorithms* are algorithms designed to run on hardware consisting of many interconnected processors. Pieces of a distributed algorithm run concurrently and independently, each with only a limited amount of information. The algorithms are supposed to work correctly, even if the individual processors and communication channels operate at different speeds and some components fail.”

Distributed Systems

- From “Distributed computing”
 - H. Attiya, J. Welch
 - Wiley, ISBN 978-0-471-45324-6
- “A *distributed system* is a collection of individual computing devices that can communicate with each other. This very general definition encompasses a wide range of modern-day computer systems, ranging from a VLSI chip, to a tightly-coupled shared memory multiprocessor, to a local area cluster of workstations, to the Internet.”

Distributed Systems

- From “Distributed algorithms”
 - G. Tel
 - Cambridge U. Press, ISBN 0-521-47069-2
- “By *distributed system* we mean all computer applications where several computers or processors cooperate in some way. This definition includes wide-area computer communication networks, but also local-area networks, multiprocessor computers in which each processor has its own control unit, and systems of cooperating processes.”

Distributed Systems

- From “Distributed systems”
 - A. S. Tanenbaum, M. Van Steen
 - Pearson, ISBN 0-13-239227-5
- “A *distributed system* is a collection of independent computers that appears to its users as a single system.”

Distributed Systems

- Good reasons for using them
 - Information exchange
 - Resource sharing
 - Increased reliability (replication)
 - Increased performance (parallelization)
- Wide range of applications
 - Telecommunications
 - Distributed information processing
 - Scientific computing
- Real life examples
 - Telephone systems
 - Airline reservation systems
 - Banking systems

Classification

- Models
 - Interprocess communication
 - Timing assumption
 - Failures
- Uncertainty and independence
 - Unknown number of processors, topology
 - Independent inputs at different locations
 - Programs started at different times, different speeds
 - Uncertain message delivery time

Inter-process Communication

- How process communicate?
- Commonly
 - Sending point-to-point or broadcast messages
 - Remote procedure calls
 - Shared memory
- Computer networks
 - messages

Timing Model

- Synchronous
 - Processors and communication work at perfect lock-step synchrony
- Asynchronous
 - Processors and communication take arbitrary time
- Partially synchronous
 - Wide range of assumptions about timing

Distributed System Models

- Synchronous model
 - Message delay is bounded and the bound is known.
 - E.g., delivery before next tick of a global clock.
 - Simplifies distributed algorithms
 - "learn just by watching the clock"
 - absence of a message conveys information.
- Asynchronous model
 - Message delays are finite, but unbounded/unknown
 - More realistic/general than synchronous model.
 - "Beware of any model with stronger assumptions." - Burrows
 - Strictly harder/weaker than synchronous model.
 - Consensus is not always possible

Failures

- Distributed systems without failures
 - ... piece of cake
- Failures are always possible
 - Harder problem solutions, sometimes impossible
 - f -failure resilient algorithms
 - wait-free algorithms
- Common type of failures
 - Process stop or Byzantine failures
 - Duplication, loss or reordering of messages

Complexity Measures

- Complexity measures
 - Time
 - Message complexity
- Time
 - synchronous system: rounds
 - asynchronous systems: need external observer
 - partially synchronous: there are time bounds
- Messages
 - number of messages
 - sometime also size (number of bits sent)

Distributed System

Thanks!