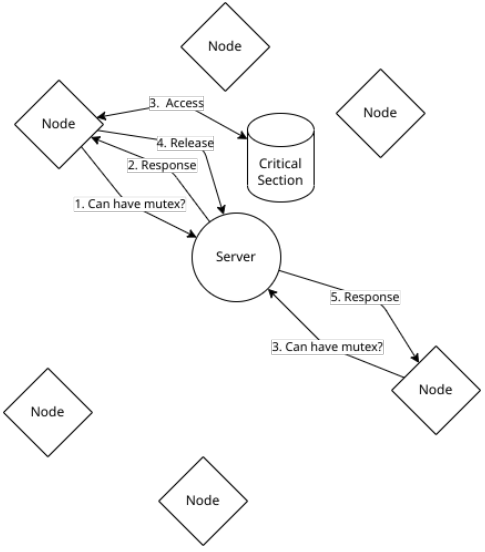
Distributed Mutual Exclusion

What is a mutex? Kinda a Lock for distributed systems ...
In a distributed system a mutex is for locking a shared resource in a network, traditionally in a program, a lock would be enough to handle concurrent writes/reads, but when the communication is expensive, locking the resource is harder to do.

Mutex Algorithms

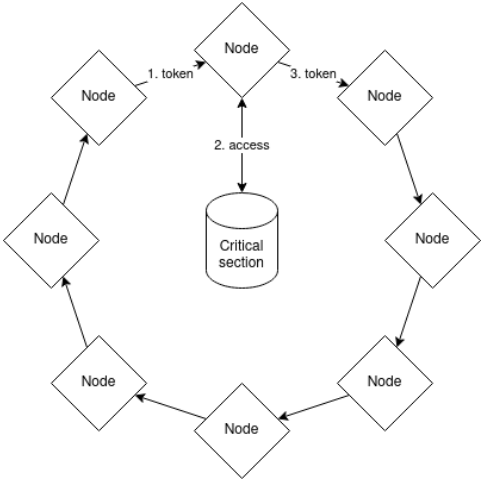
- Centralized Algorithm
- Token Ring Algorithm
- Ricart and Agrawala's algorithm
- Maekawas algorithm

Centralized algortihm



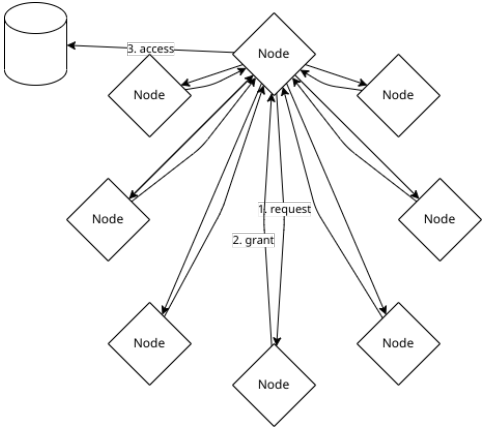
All three nodes highlighted are points of failure.

Token Ring Algorithm



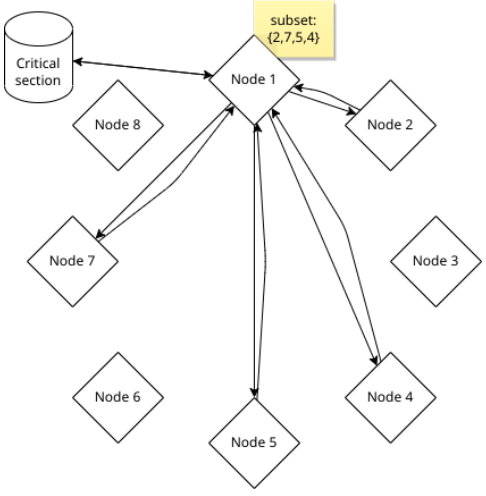
All nodes are points of failure at any given time.

Ricart and Agrawala's algorithm



All requests are multicast.
There is a 4th step where a release is multicast to all nodes as well.
All nodes are points of failure at any given time.

Maekawas algorithm



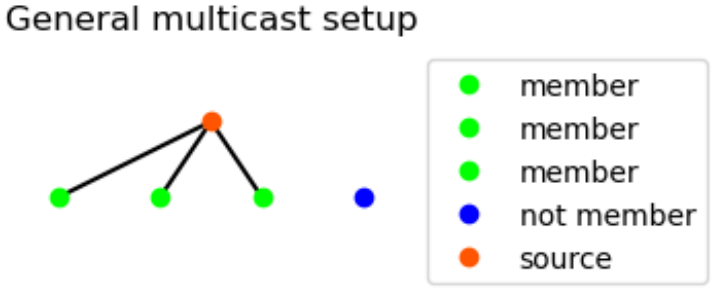
Very similar to Ricart and Agrawala, but only nodes in the subset failing will result in a particular node to fail. If you're a bit smart about choosing the subsets this can be minimized.

Performance

Bandwidth means Entry and Exit potential. This means how many times an acces can happen per messages.
Token ring is infinite because the node itself chooses when to send the token.
Ricart and Agrawala's algorithm has a bandwidth of 1+n-1 if it can do multicast in hardware, otherwise its n-1+n-1, as it will now have to unicast to everyone.

Algorithm	Entry	Exit	sync	Bandwidth
Centralized	2	1	2	2 / 1
Token Ring	a:n/2 w:n-1	1	a:n/2 w:n-1	∞ / 1
R + A	2	2	1	1+n-1 or n-1+n-1 / n-1
maekawa	2	1	2	$4\sqrt{n}-2$ / $2\sqrt{n}-1$

Multicast/Group Communication



IP Multicast - Hardware Support

Uses a protocol called IGMP
If there's hardware support, the sender transmits one message and the intermediary devices will determine how many devices it will transmit the message to.
If there is not hardware support, then the sender will have to unicast many messages at once to its receivers instead.

IP Multicast - Problems

Out of order delivery, this can happen due to changes in routing. Consider a network with a slow and a fast intermediary node as two independant routes, if the sender sends a message through the slow node to the reciver followed by a message through the fast node, then the receiver might receive the second message before the first.

IP Reliable Multicast

- Trading efficiency for reliability
- ACKs - increases time complexity to $O(n^2)$

FIFO! First In - First Out Multicast

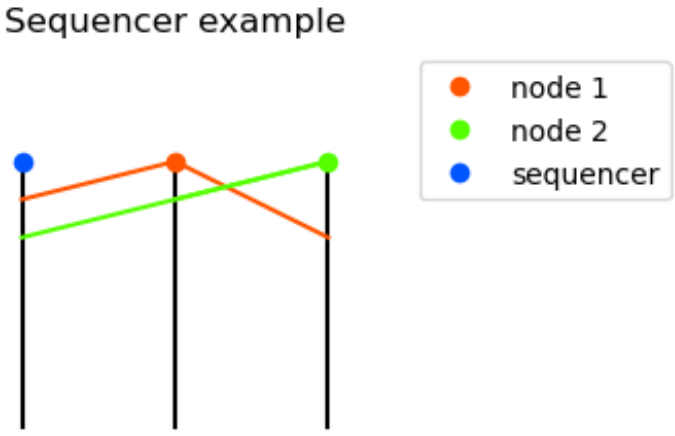
Respects sequence numbers

Total Ordering

Two requirements must be met:

- Messages should arrive in the same order as they were sent
-

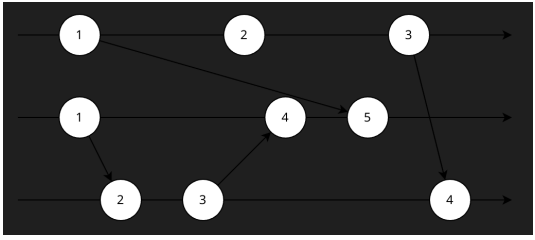
Using a Sequencer

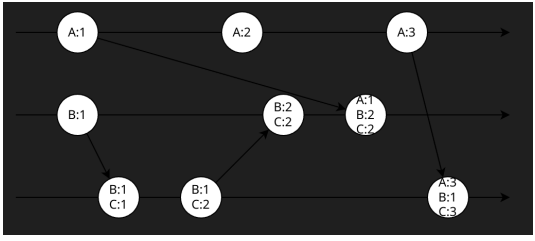


Causal Ordering

generally, causal ordering means that the events in a network does not depend on a future event, this is usually realized using lamport clocks.

Clocks for ordering





Consensus Protocols

The idea is to make sure processes agree
Synchronous Consensus Algorithm
Byzantine Generals Algorithm
Phase-King Algorithm

Byzantine faults

The armies example, one half retreats, one attacks.
Two different descisions was made, this is an error
Byzantine failures are by extension when the whole system fails due to a byzantine fault.

Correct Processes

A correct process is a process that is operating as expected. Closely tied with byzantine faults, if a process has a byzantine fault, then it is no longer a correct process.

Synchronous Consensus Algorithm

f-resilience, f processes may fail

- initially multicast your value
- do f+1 rounds
- in each round, process received messages, record how many times each unique value is received
- multicast again

Byzantine Generals

> you have commanders and lieutenants, its kinda like master and slaves in other systems, the commander says a value, and the lieutenants take this value as the correct value.
> In this system, if the commander is not correct, it results in a failure.
> when the value is received from the commander, the lieutenants multicast their received value and id, then the majority value is chosen at each process.

Phase-King algorithm

> generals only work for f = 1, and can be optimized f+1 phases

> in each phase, all processes start by transmitting their value, if the most frequest received value is from more than half of the processes, use that value

> the king is elected, process with id k is elected at phase k

> the purpose of the king is to send a tie breaker if two processes share the same count of values.

Replication and Consistency

Goal of replication:

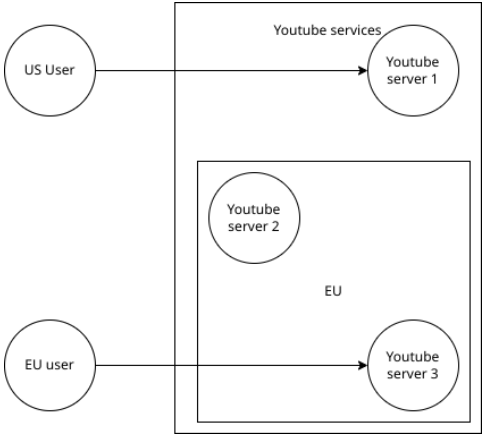
> To tolerate failures

> High availability

> More affordable to scale out

> caching is replication

Replication



> each server has videos cached

> if server 3 goes down, the EU user can use server 2

Passive Replication

> One primary node, all other nodes are backup

Consistency

Youtube: like count is consistent while views are relaxed consistency

What if user A and user B of a bank system adds money at the same time? all nodes has to agree on one value

Sequential consistency

Follow causality,

Distributed Storage

Cheaper/more efficient to scale out storage rather than scale up.
imagine a large database which gets overloaded with queries, its a better idea to have multiple servers handle the load.
Imagine if youtube had all videos on one system, it would perform better if you had two systems and every second video is on the other system.

Big Data Storage

Often one system can't handle the amount of data stored

GFS Filesystem

A filesystem designed to have direct access to files across a cluster of distributed storage nodes.

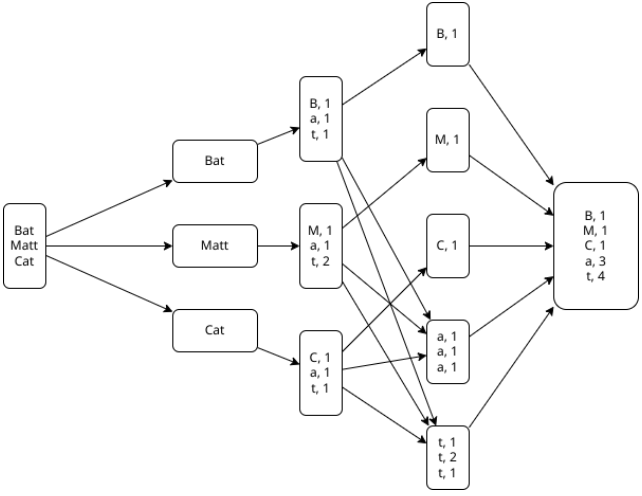
Big Data Analytics

Why? Lots of unstructured data could be processed
Not suited for traditional databases

Map Reduce

It is a method to take large chunks of data, split them and process them in parallel on distributed nodes, and finally aggregating them.
Functional programming style data processing:
Map: maps a key to generate some value
Reduce: any equivalent key is reduced together with similar keys to single values.

Map Reduce


Bat
Matt
Cat
Bat
Matt
Cat
B, 1
a, 1
t, 1
M, 1
a, 1
t, 2
C, 1
a, 1
t, 1
B, 1
M, 1
C, 1
a, 1
a, 1
t, 1
t, 1
t, 2
t, 1
B, 1
M, 1
C, 1
a, 3
t, 4

Fault Tolerance

workers are pinged by master node (first in previous diagram)
master is a single point of failure

Spark (RDD)

Blockchains

Blockchains are Peer-To-Peer Networks Hashing

> Compare a hash representing data instead of comparing data

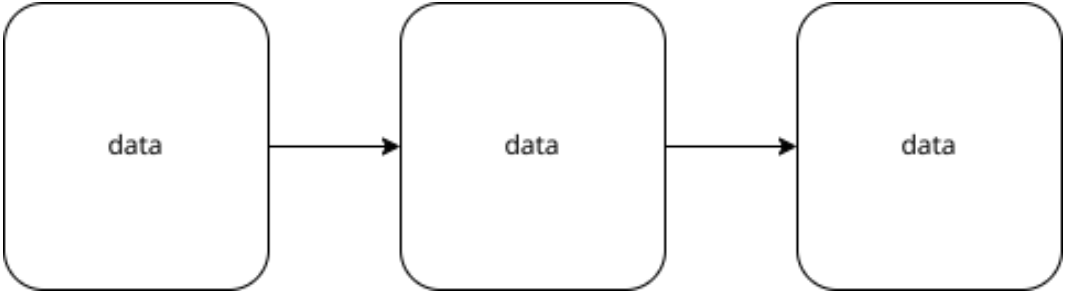
> Make hashes for portions of data if its too large

The Three Algorithms

generateKeys
> Generate private and public keys
sign
> Compute a hash based on the secret key
verify
> Check if the message and hash matches based on the public key
Example: passwordless ssh

The Blockchain

each block is identified by a hash



Peer-to-Peer Networking

Overlay networks and underlying network

Three types of peer-to-peer networks

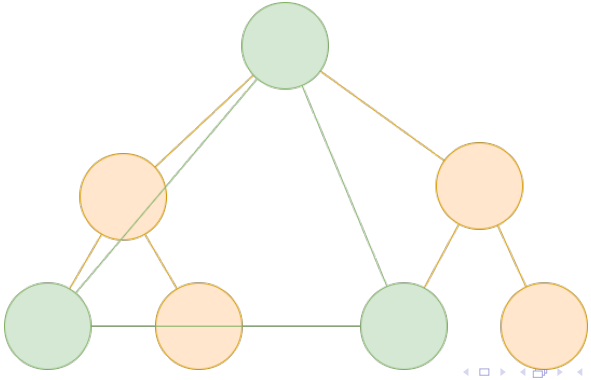
> Unstructured

> Structured

> Centralized

Overlay Networks

This network is like an abstraction of the underlying architecture.
Green: overlay, Yellow underlying network



Unstructured P2P Networks

No central server
No structure in the overlay network
Devices come and go from the network very frequently
very slow search due to flooding

Structured P2P Networks

Can be structured according to rings.
in a ring, a node would maintain pointers to the next node and previous node
if a node wants to join, it'll first contact an arbitrary node, that will contact around the ring until the previous node can tell the new node who its next is, connection between the new and next node is established and finally the new node can join.

Centralized P2P Networks

Structured around a centralized server

Centralized server has no data, it will just return the physically closest other client that has the data

Advantages/Disadvantages

Advantages

> Inherently scalable

> Inherently Fault Tolerant

Disadvantages

> Security

> Backups are hard

Internet of Things and Routing

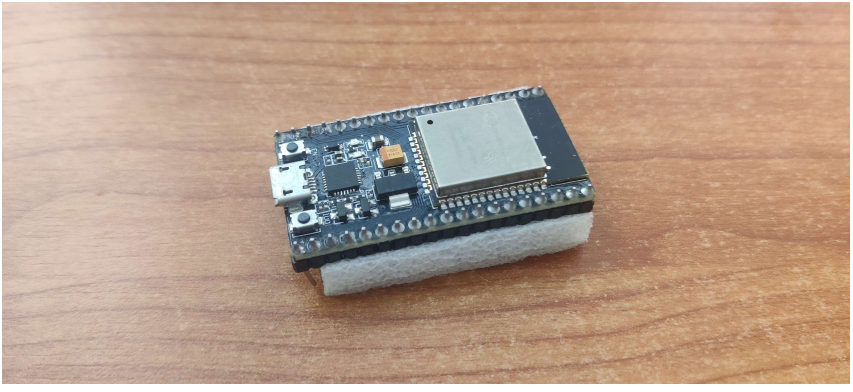
Terminology:

> Embedded Systems, WSN, CPS, IoT

Low power, Real time, etc...

Wireless Sensor Networks

monitoring
Example: P7 Project
First draft project: Deep Sleep and energy saving



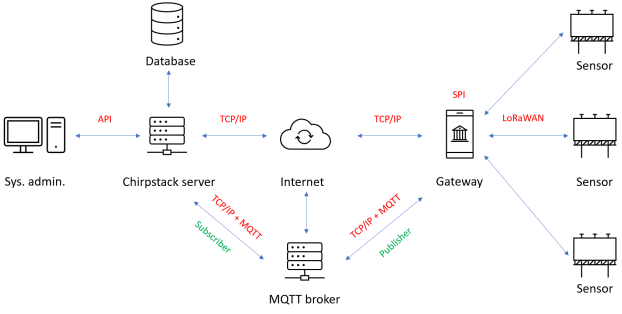
Typical Protocols in WSN/IoT

WiFi WLAN

Bluetooth PAN

Others? ZigBee, LoRA, CSMA, ...

Example: P1 LoRA wireless sensors



CSMA

MAC Layer protocol

You can compare this protocol with TDMA (Time Division Multiple Access) where the channel is split into time slots for each user, while CSMA listens for concurrent transmissions before transmitting.

> Many devices waiting on a channel to be ready is prone to collisions

> Collision Avoidance

Route Discovery DSR

Flooding