

**chess**

engineering the future



**Grootschalige draadloze sensornetwerken:**

**Onzin of realiteit?**

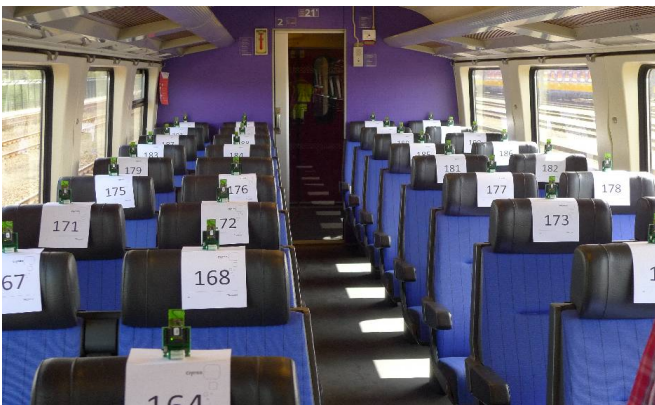
# Het is realiteit....



In kassen: sense & control



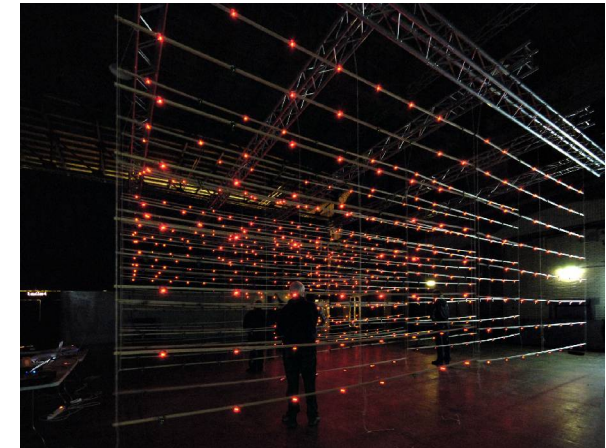
Traceren van goederen:  
actieve labels



Stoelreservering in de trein:  
service

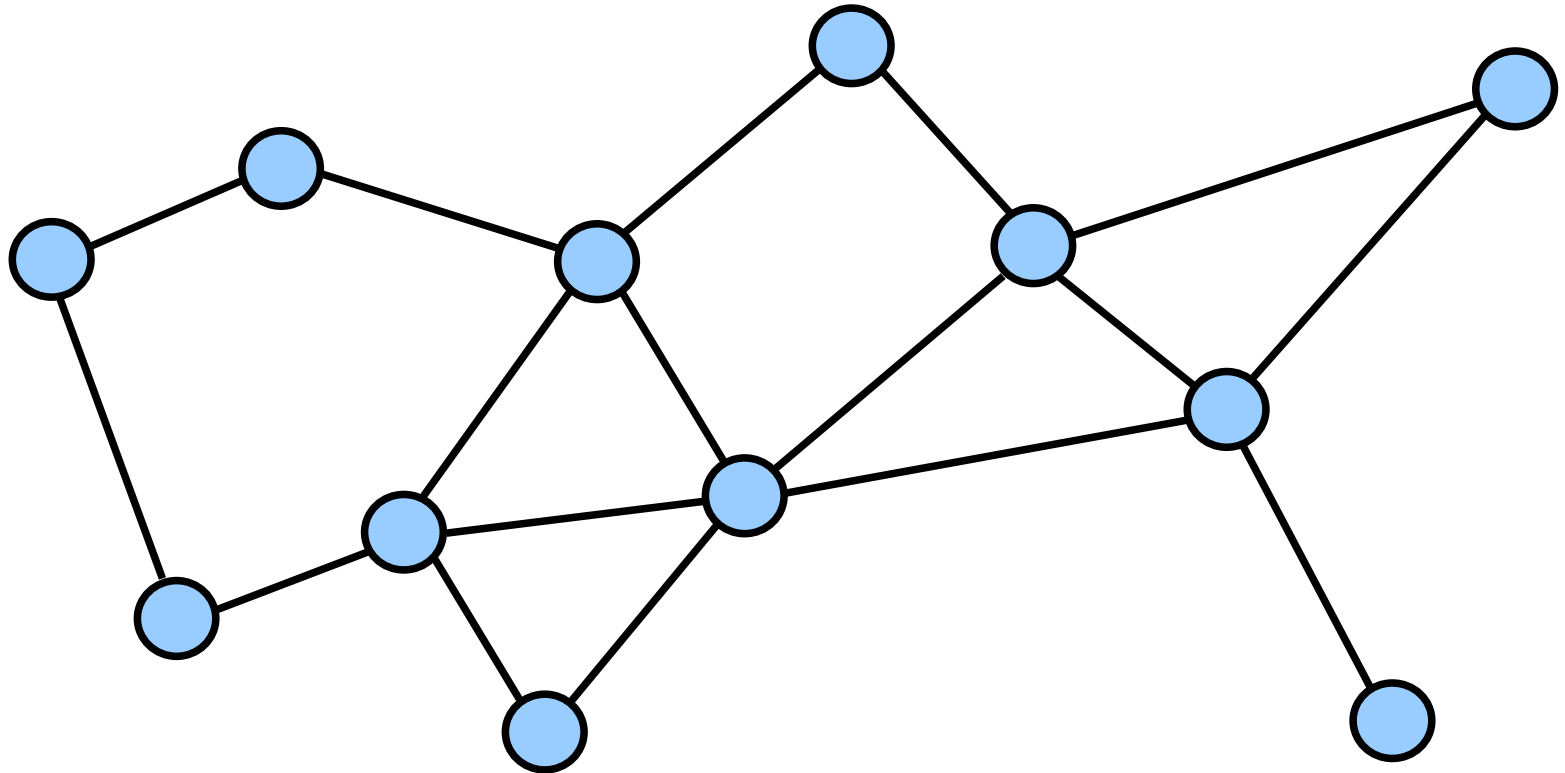


Gezondheidszorg:  
ondersteuning



ALwEN research project:  
1000 Node experiment

# De theorie



# Inspiratie uit de natuur

Broadcast or Newscast  
message delivery



Audience listens  
and take local action



Firefly



Synchronous flashing

# Meer inspiratie...



Ant colonies & Cell biology:  
local computation, global effect

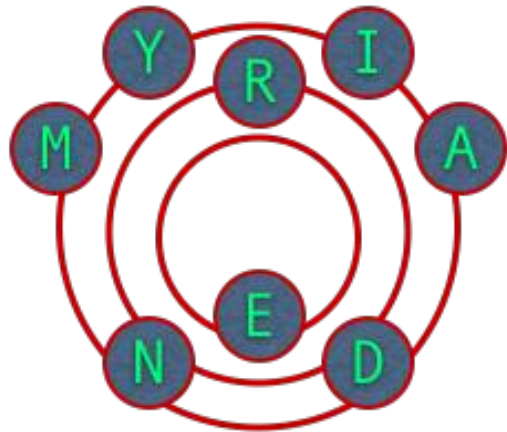


Gossip message exchange



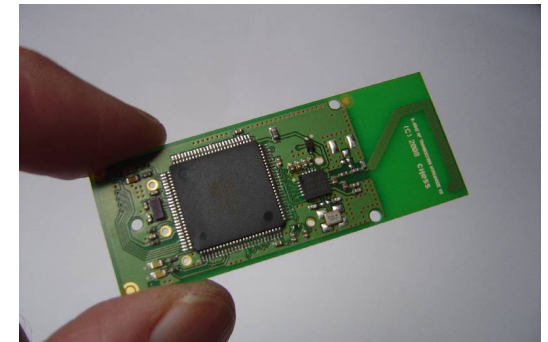
social interaction

# MyriaNed

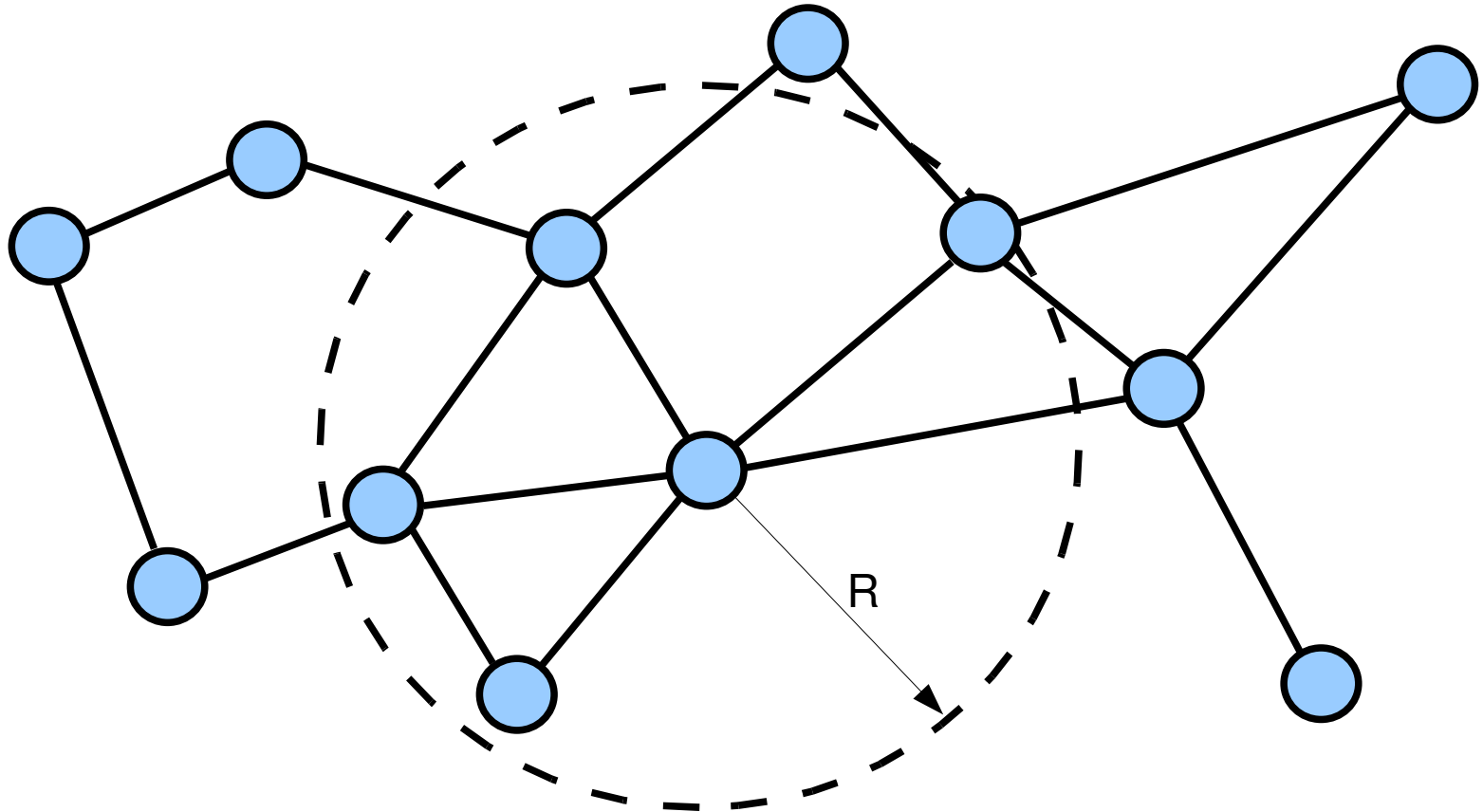


## Eigenschappen:

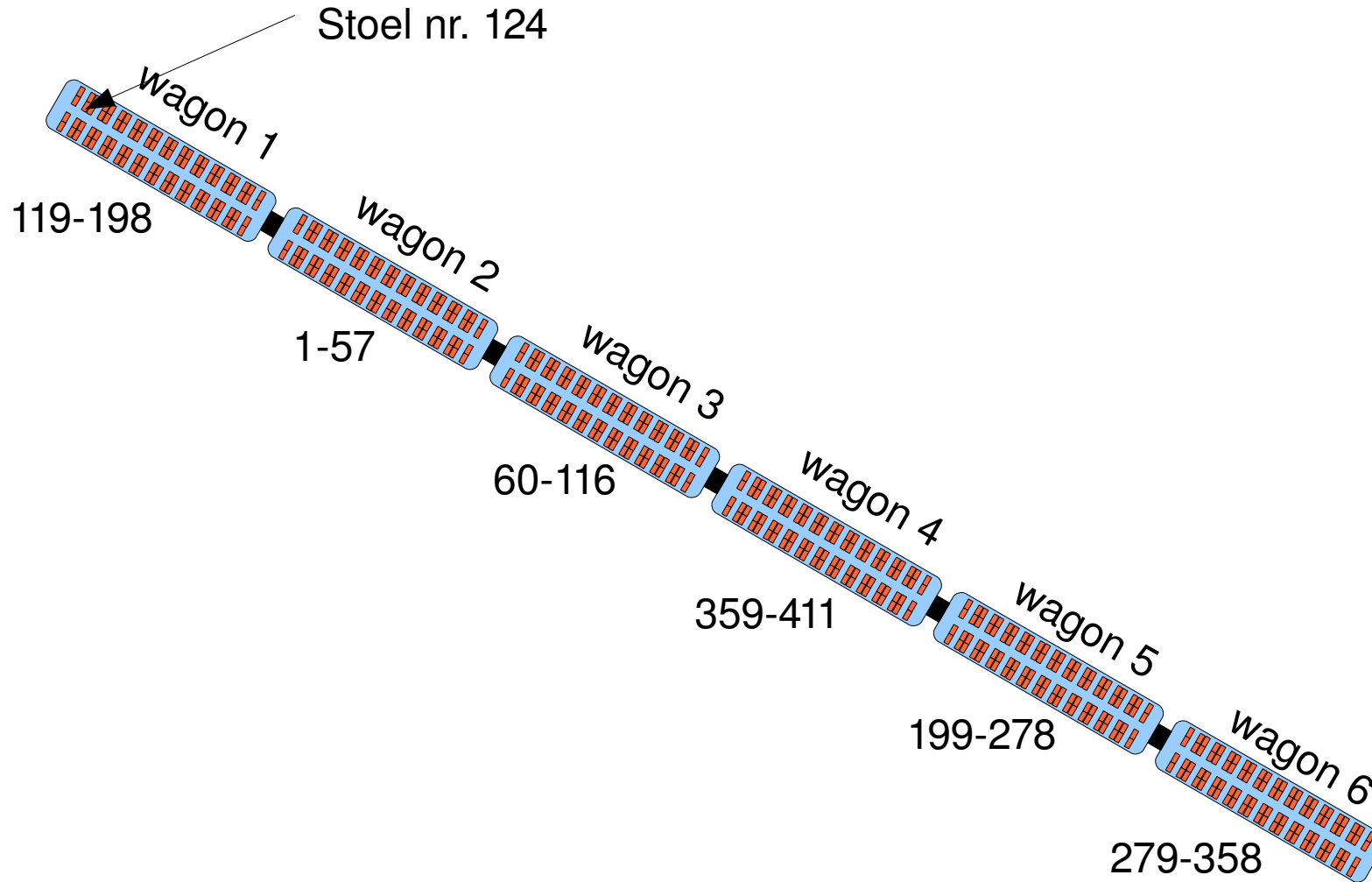
- Ad hoc
- Zelforganiserend
- Schaalvrij
- Gebaseerd op roddelen
- Zeer laag energieverbruik
- Klein en compact
- Elegant
- Robuust



# Unit-disc radio model?

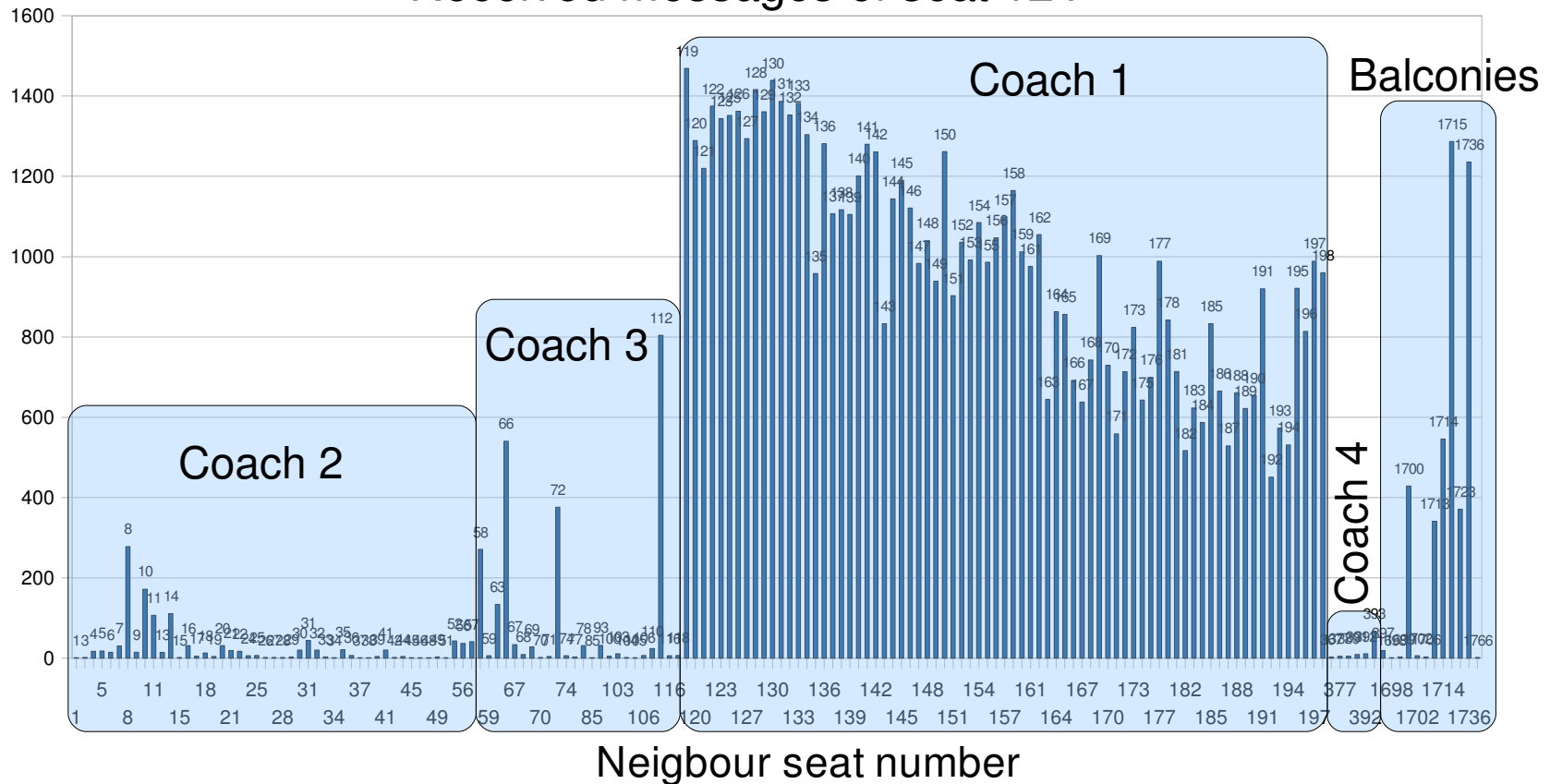


# Zitplaatsreservering in de trein

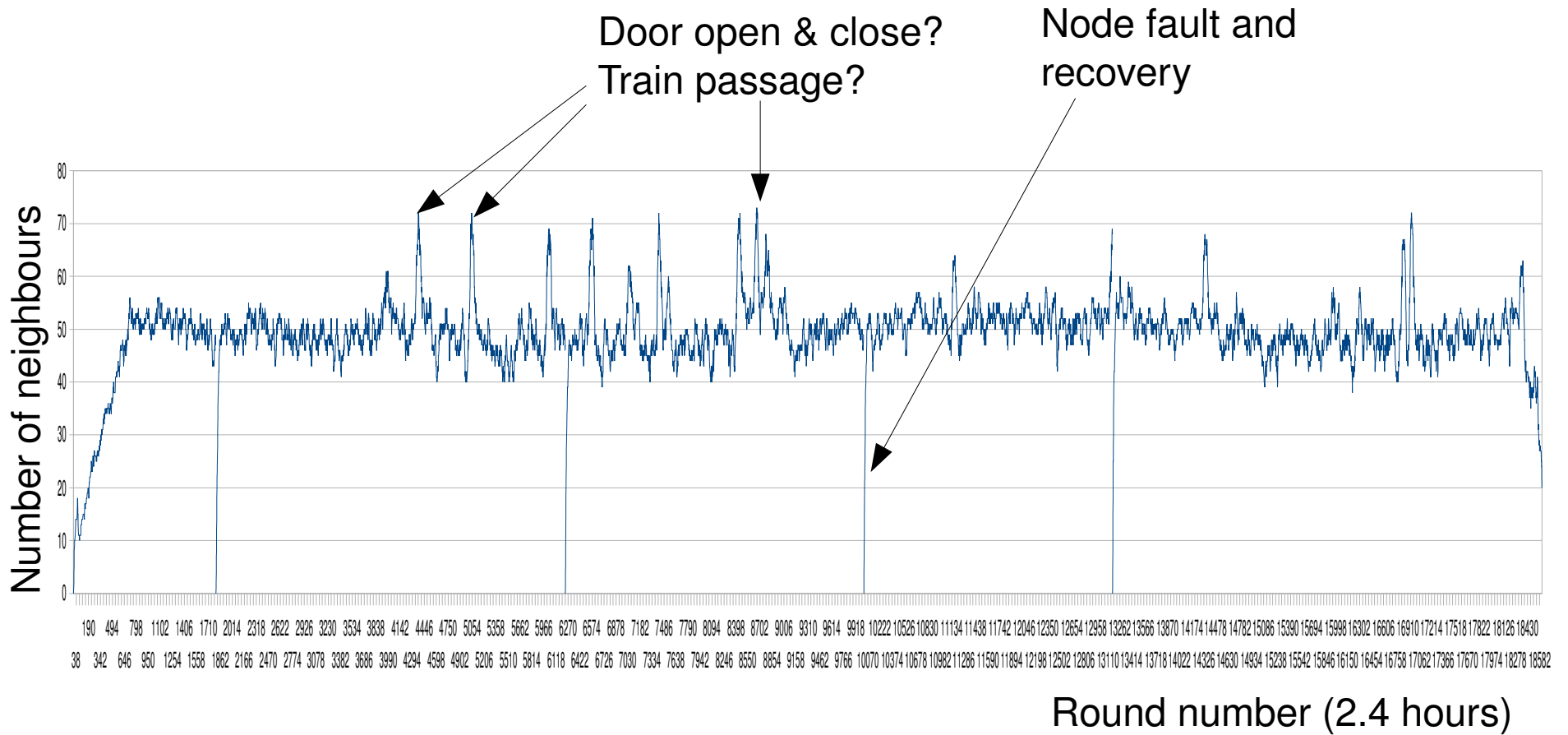


# Unit-disc radio model.....

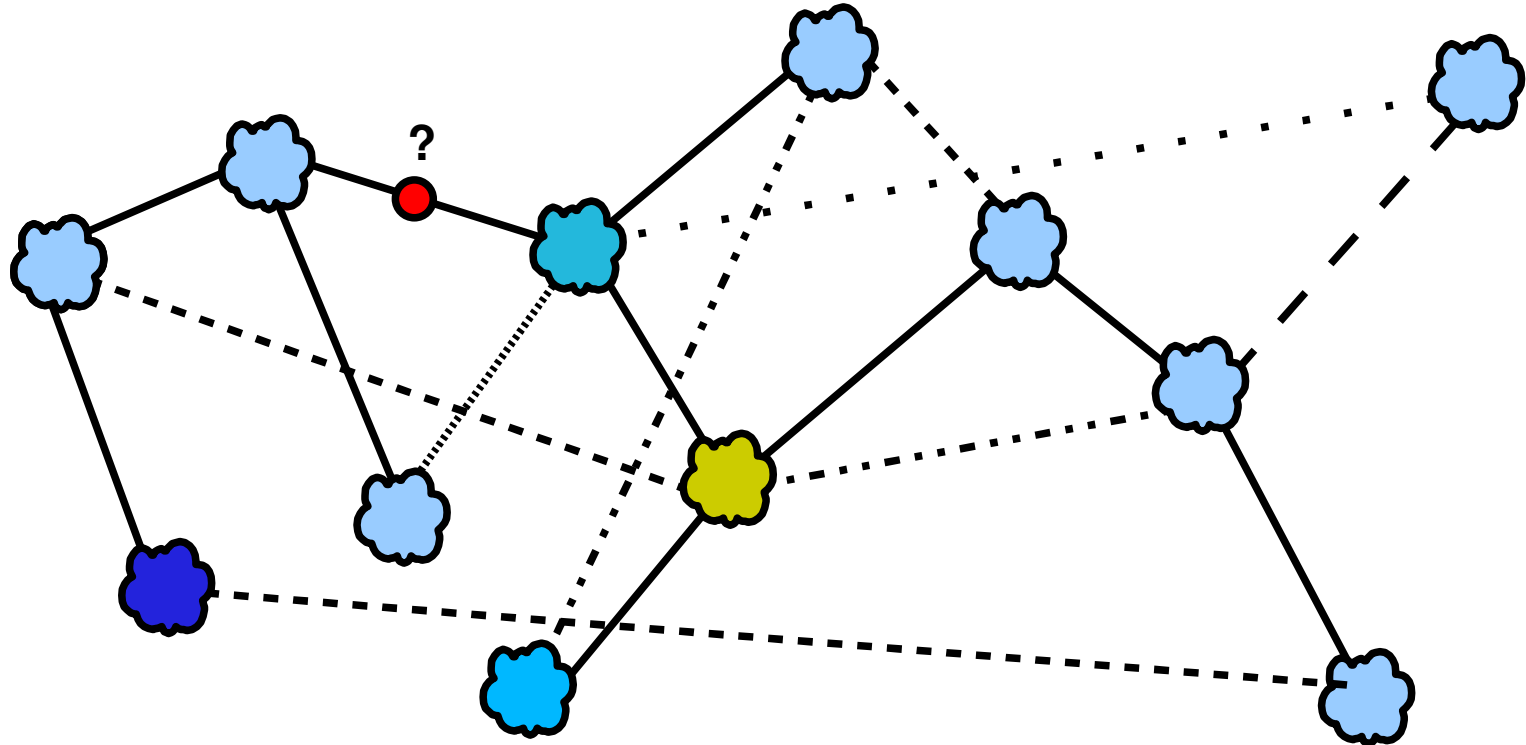
Received messages of seat 124



# Aantal buren van stoel 124



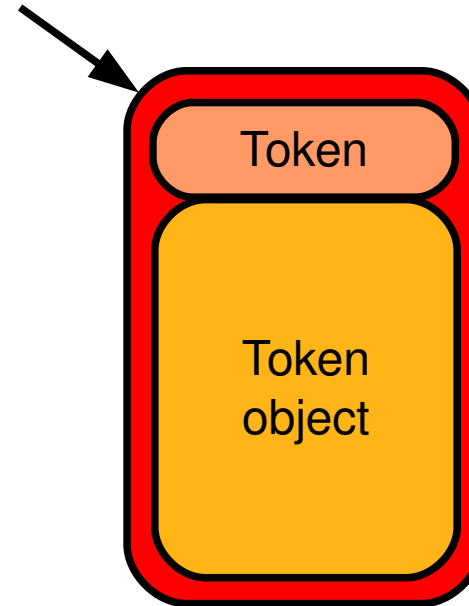
# Organische structuren?



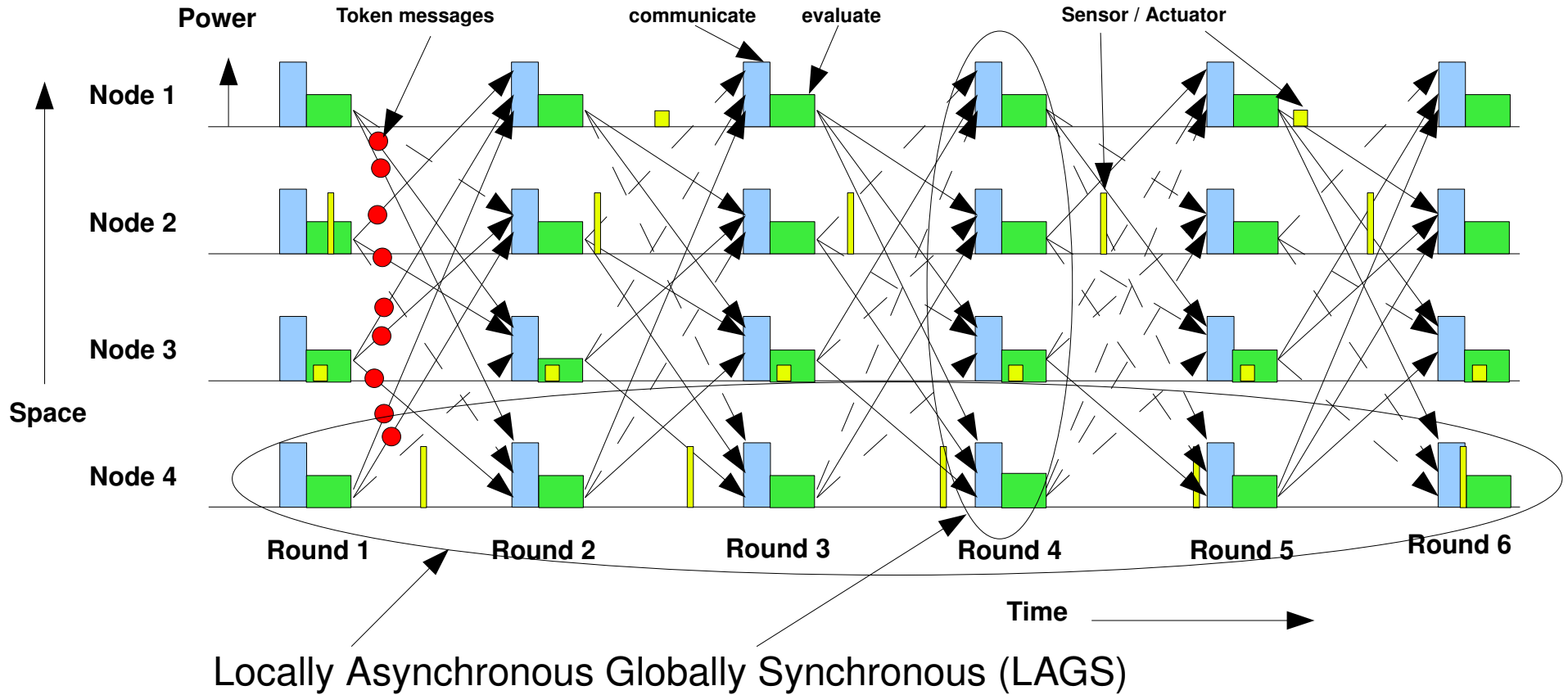
# Distributed Token Machines



Token Message

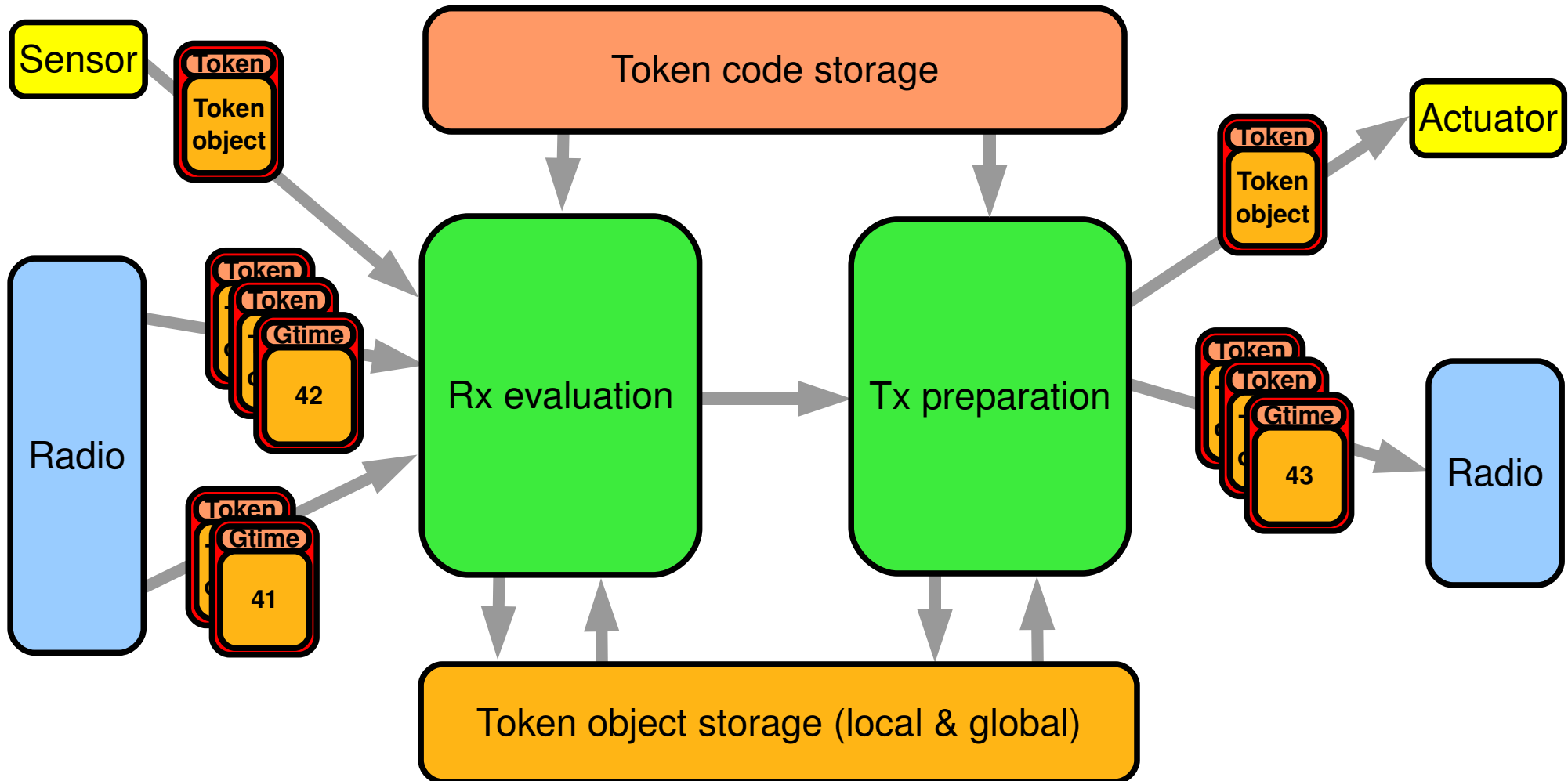


# Network execution model (Distributed Token Machine)



# MyriaCore Token Machine

## Data flow & paths



# TML applicatie

```
void tmInit(void) {...} // Initialize all tokens

void tmEvalRxMsg(gMacMessageT* pRxMsg) { // evaluate each message
    tokenImplicitEvalGmacSynchronize(pRxMsg);
    tokenImplicitEvalGmacStrategy(pRxMsg);
    tokenImplicitEvalNeighbourList(pRxMsg);
    while (tokens in pRxMsg) {
        if (GT_token) tokenEvalGlobalTime(TokenObject);
        if (GS_token) tokenEvalGossip(TokenObject);
        if (GD_token) tokenEvalGradient(TokenObject);
    }
}

void tmPrepareTxMsg() { // Prepare the message to transmit
    pTokenMsg = 0;
    tokenImplicitDoGmacSynchronize();
    tokenImplicitDoGmacStrategy();
    tokenImplicitPrapareNeighbourList();
    pTokenMsg = tokenPrepareGlobalTime(pTokenMsg);
    pTokenMsg = tokenPrepareGossip(pTokenMsg);
    pTokenMsg = tokenPrepareGradient(pTokenMsg);
}
```

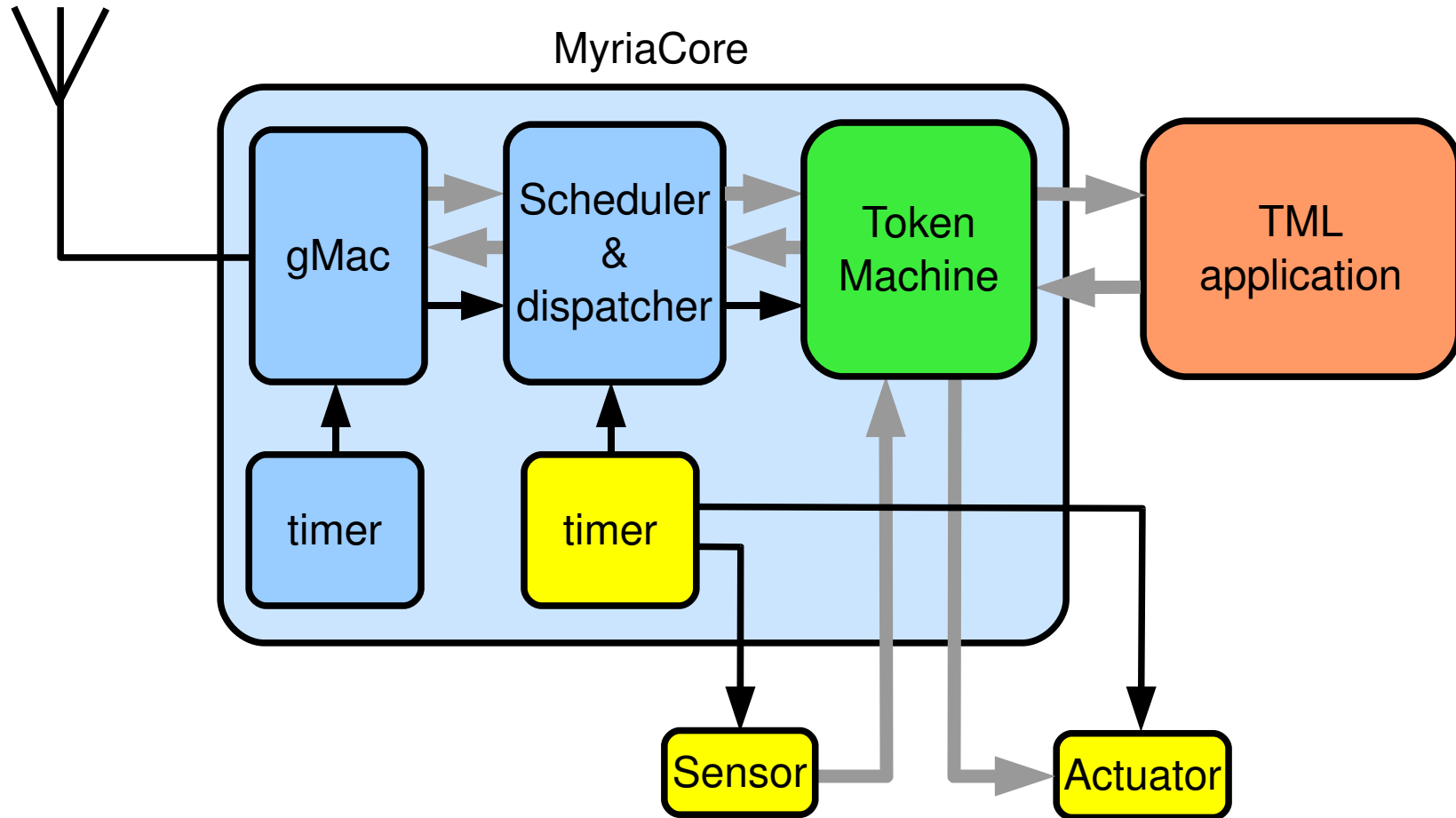
# “global time” Token (implementatie) **chess**

engineering the future

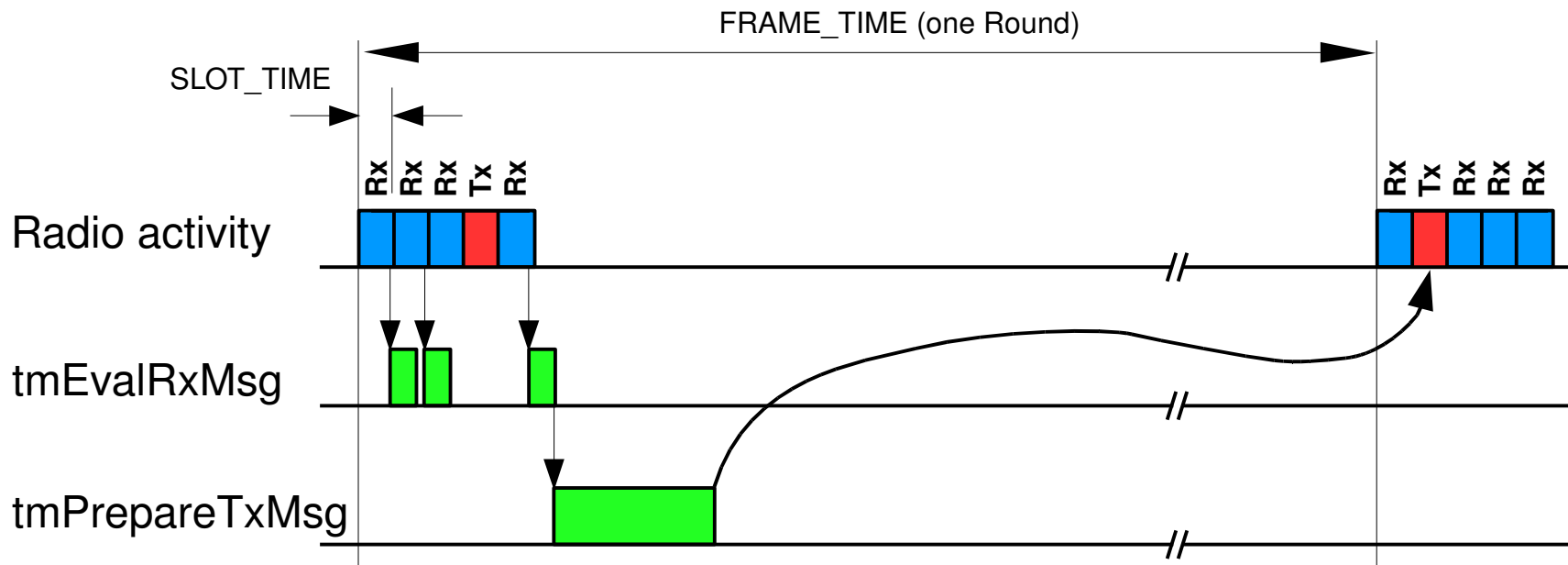
```
unsigned int myGlobalTime; // global token storage
```

```
tokenInitGlobalTime() {  
    myGlobalTime = 0; // initialisation value  
}  
  
tokenEvalGlobalTime(globalTime) {  
    if (greater(globalTime, myGlobalTime))  
        myGlobalTime = globalTime; // Adjust my global time value  
}  
  
tokenPrepareGlobalTime() {  
    if (myGlobalTime == 0)  
        ; // not yet heard anyone. Do nothing!  
    else if (myGlobalTime == MAX_INT)  
        myGlobalTime = 1; // wrap around  
    else  
        myGlobalTime++; // advance time  
    return myGlobalTime;  
}
```

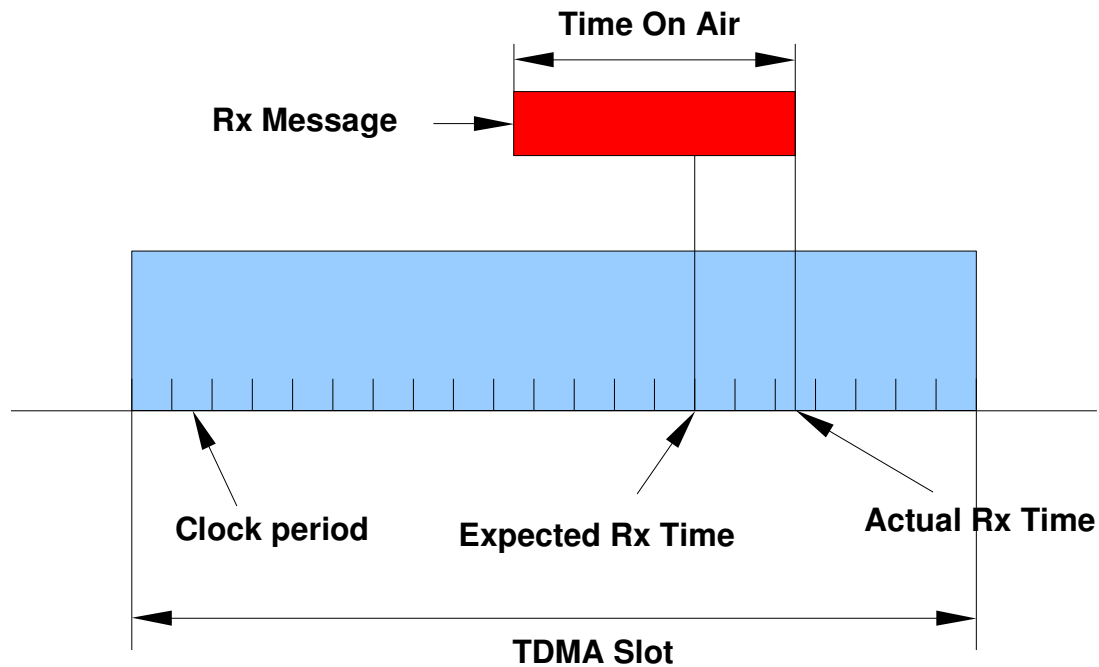
# MyriaCore, gMac & Timing



# gMac TDMA timing



# gMac synchronisatie



# MyriaNed R&D

chess

engineering the future

## DevLab leden:

- Almende
- Chess
- Connect
- IYC
- Luminis Live
- Mediatronix
- NanoSense
- NBG
- Salland Electronics
- Van Mierlo
- Vitelec

## Onderzoeksprojecten:

- ALwEN (pointOne)
- CCF2 (Pieken in de Delta)
- Metaverse1 (ITEA2)
- PLEISTER (STW)
- QUASIMODO (FP7)
- STORM (Pieken in de Delta)



## Zie verder:

<http://www.devlab.nl>

<http://wsn.chess.nl>

## Onderzoekspartners:

- ESI
- Holst Centre
- Philips Research
- Roessingh Research
- RWTH Aken
- TU Delft
- TU Eindhoven
- Universiteit Aalborg
- Universiteit Nijmegen
- Universiteit Saarland
- Universiteit Twente
- VU Amsterdam
- en anderen

**chess**

engineering the future

Lichtfabriekplein 1

2031 TE HAARLEM

The Netherlands

T +31 (0)23 514 91 49

F +31 (0)23 514 91 99

W [www.chess.nl](http://www.chess.nl) - [WSN.chess.nl](http://WSN.chess.nl)

E [info@chess.nl](mailto:info@chess.nl)

