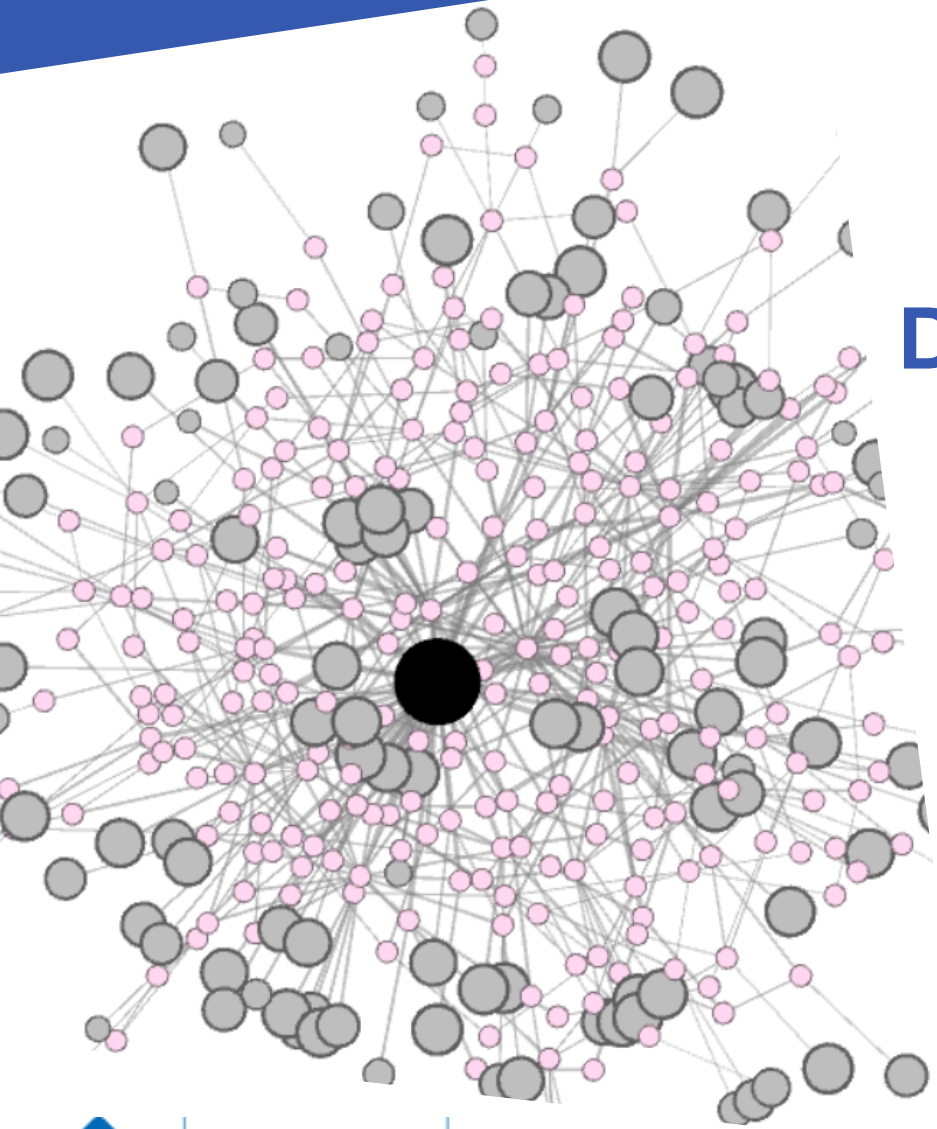
ordcot.uib.cat



Comparing Centrality Indices for Network Usage Optimization of Data Placement Policies in Fog Devices

Isaac Lera, **Carlos Guerrero**, Carlos Juiz

This work was supported by the Spanish MINECO and the European Commission (FEDER funds) throught the grant number TIN2017-88547-P

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA, INDUSTRIA Y COMPETITIVIDAD
AGENCIA ESTATAL DE INVESTIGACIÓN
FEDER
Fondo Europeo de Desarrollo Regional
UNIÓN EUROPEA
"Una manera de hacer Europa"

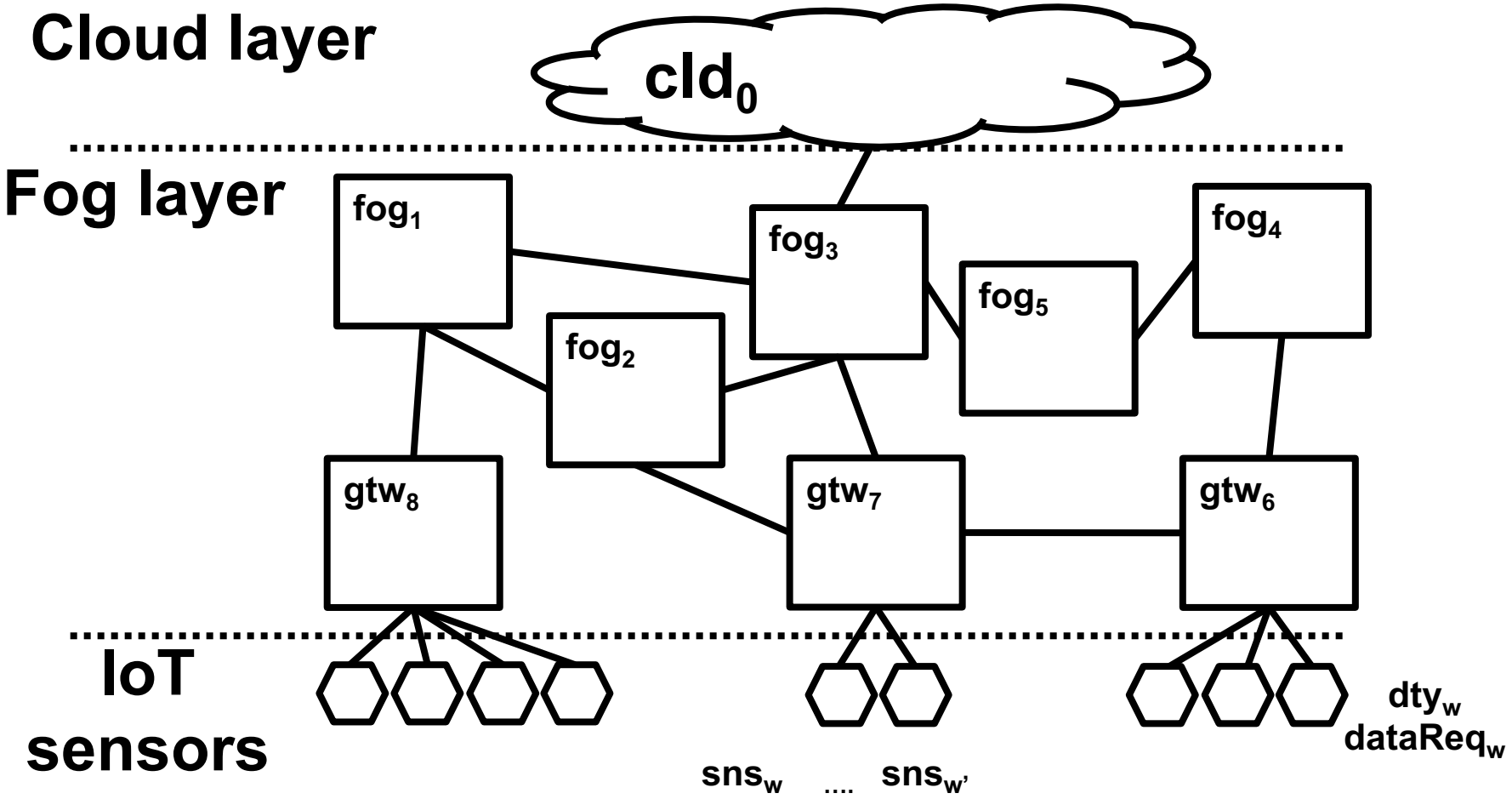
UIB Universitat de les Illes Balears
#SOM UIB

April 23rd, 2018
FMEC 2018, Barcelona

Introduction

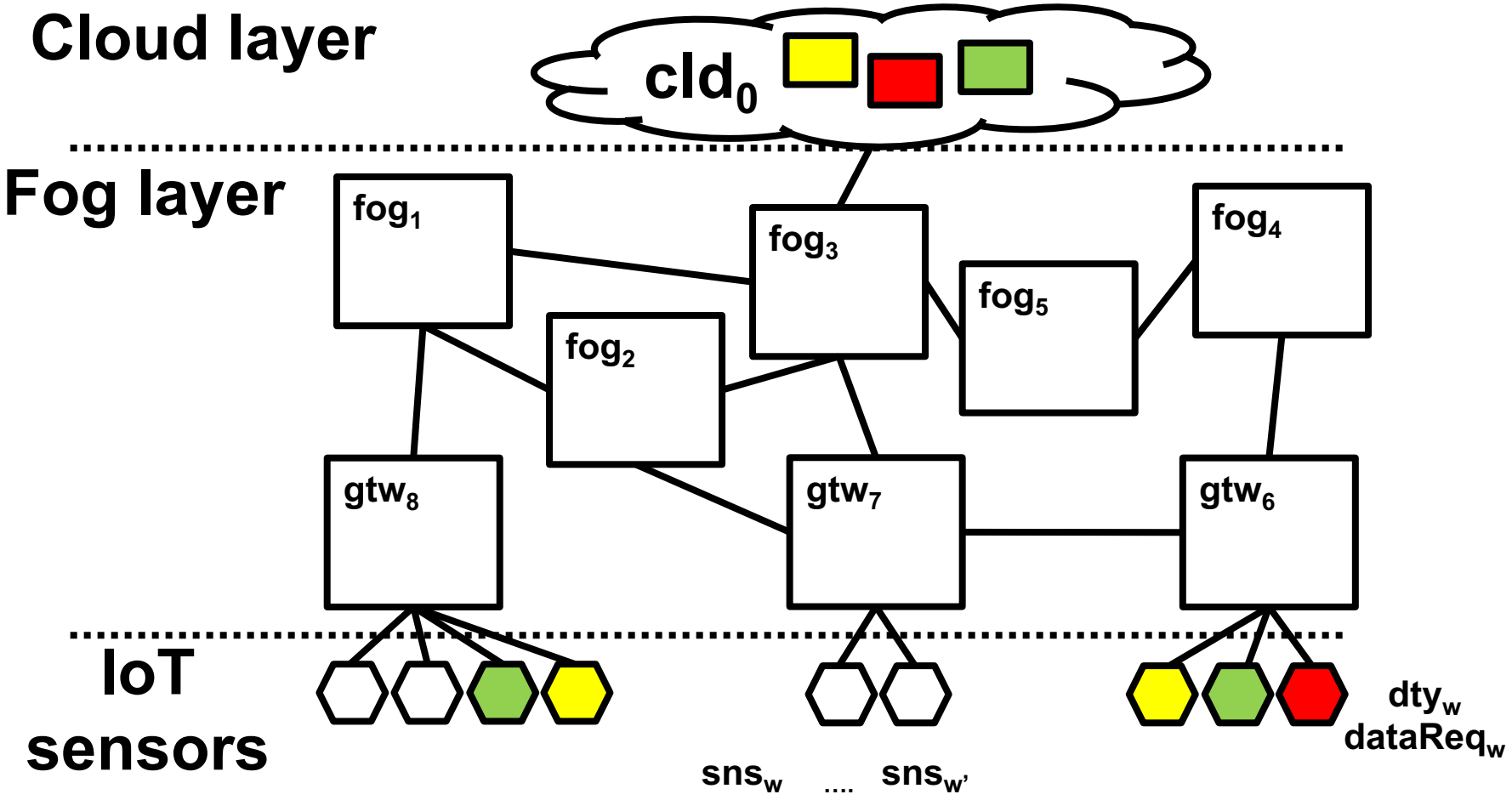
- Domain → Fog Computing
- Objective → Minimize network usage
- Decision variable → Data placement in fog devices
- Requirements → simple and scalable for large systems
- Tools → Complex Networks Metrics
- Solution → Centrality indices

Fog Architecture Model



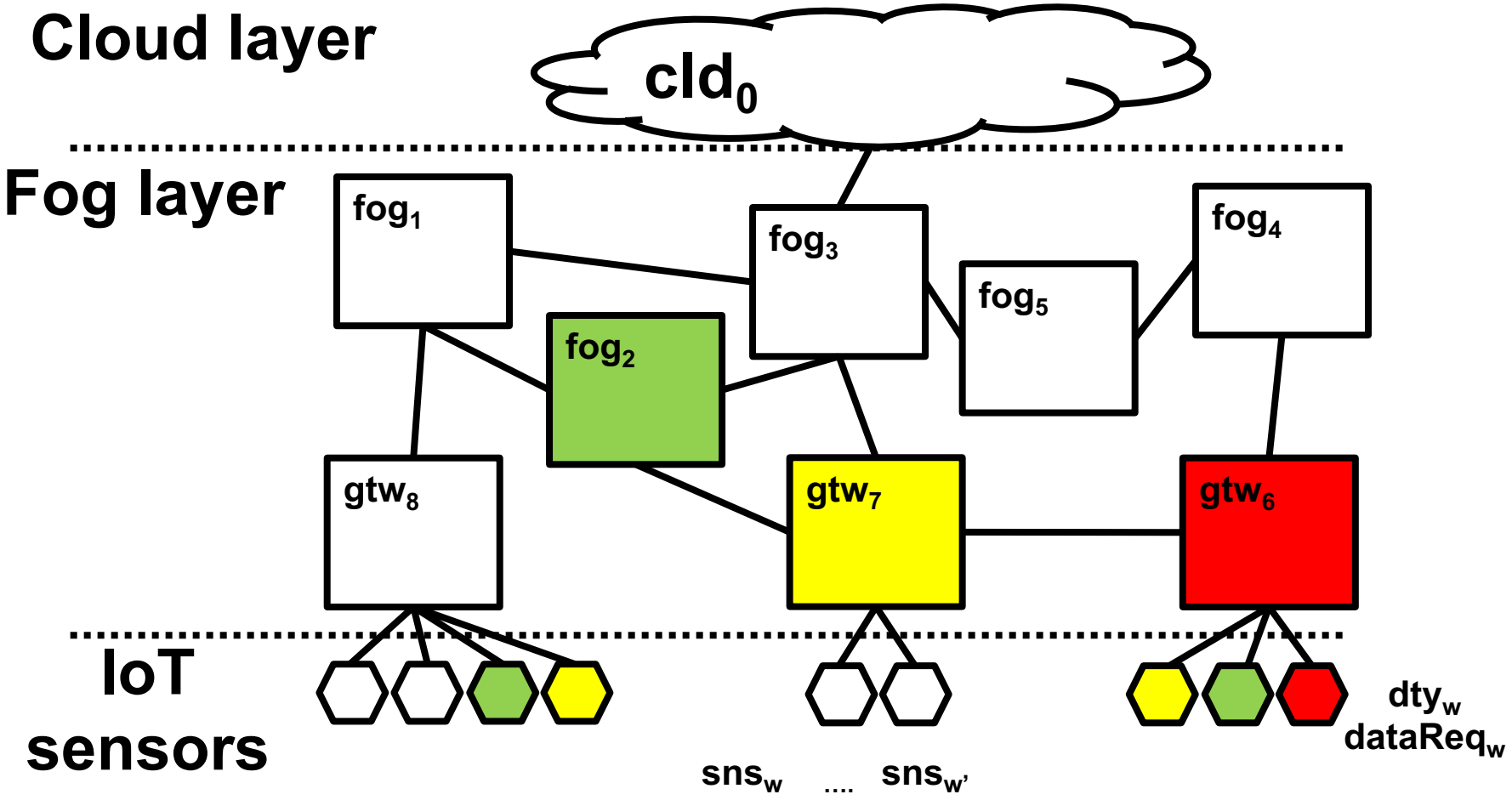
- Service placement
- Data Placement

Fog Architecture Model



- Service placement
- Data Placement

Fog Architecture Model



- Service placement
- Data Placement

Data Placement Problem

- Solution
  - Data placement matrix
  - |Data types| x |Fog devices|

$$P = \begin{cases} p_{xu} = 1 & \text{if } dty_x \text{ is stored in } fog_u \\ p_{xu} = 0 & \text{otherwise} \end{cases}$$

- Constrains
  - Data type stored in only 1 device
  - Provisioned storage smaller than fog storage

$$\sum_{u=1}^{|U|} p_{xu} = 1, \forall\ x = (1, \ldots, |X|)$$

$$\sum_{x=1}^{|X|} p_{xu} \times strReq_x \leq strCap_u, \forall\ u = (1, \ldots, |U|)$$

- Objective
  - Number of data packets transmitted

$$min \sum_{x}^{|DTY|} \sum_{u}^{|FOG|} p_{xu} \times \sum_{w}^{|SNS|} dist(sns_w, fog_u)$$

Data Placement Algorithm

(a) Model the network topology as a complex weighted network;
(b) Weight the edges with the summation of the hop distances to the sensors of a given data source type
(c) Calculate the centrality index of each node
(d) Select the node with the highest centrality index

- (d1) if not enough free storage capacity select the next highest centrality until a device with enough storage is found

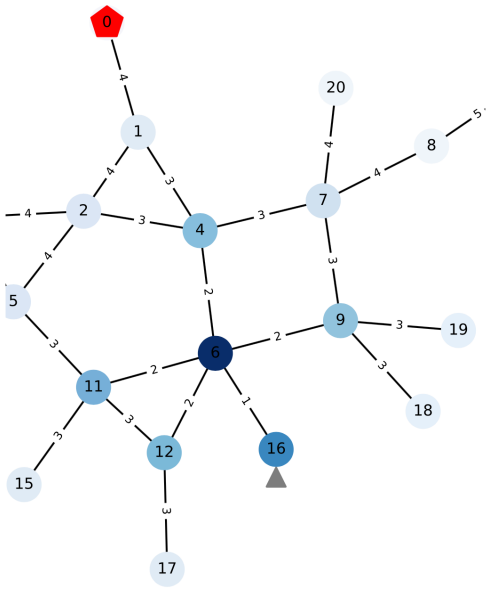
Repeat the process for each data type, ordered by the data generation rate

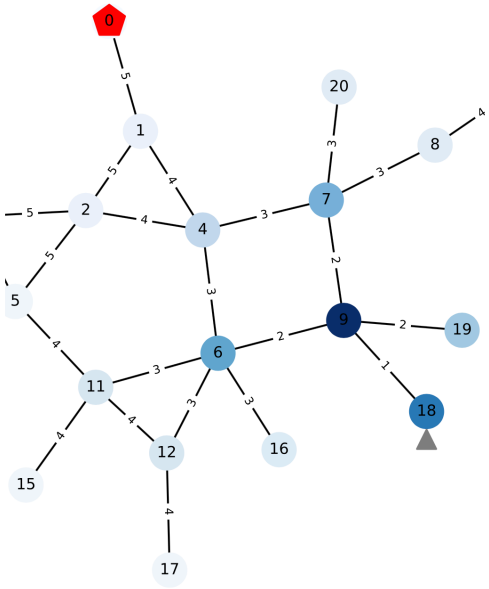
Algorithm 2 Pseudo-code of the data allocation algorithm.

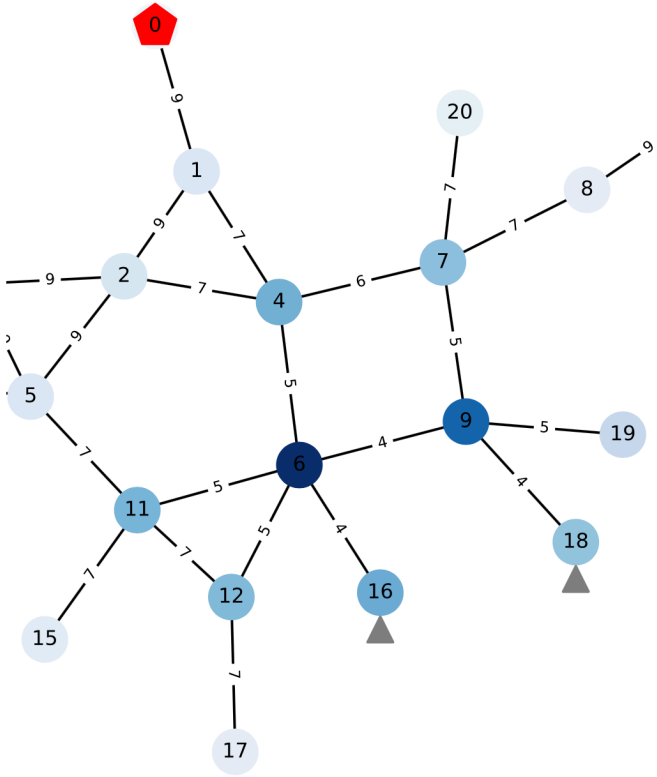
Procedure: DataAllocation()
Input: network topology, set of devices
Output: data type allocation (matrix P)

```
1: CN = EdgeWeightCalculation()
2: orderedDty = sort(|DTY|, "by generation rate")
3: for all dty_x ∈ orderedDty do
4:    nodesCentrality = centralityAglorithm(CN[dty_x])
5:    orderedNodes = sort(nodesCentrality, "by centrality")
6:    allocated = false
7:    while not allocated do
8:       fog_u = orderedNodes.next()
9:       if dty_x.dataReq_x < (fog_u.dataCap_u -
         fog_u.dataProv_u) then
10:         allocated = true
11:         fog_u.dataProv_u += dty_x.dataReq_x
12:         p_{dty_x, fog_u} = 1
13:      end if
14:   end while
15: end for
```

Data Placement Algorithm







Algorithm 1 Pseudo-code for the edge weight calculation.

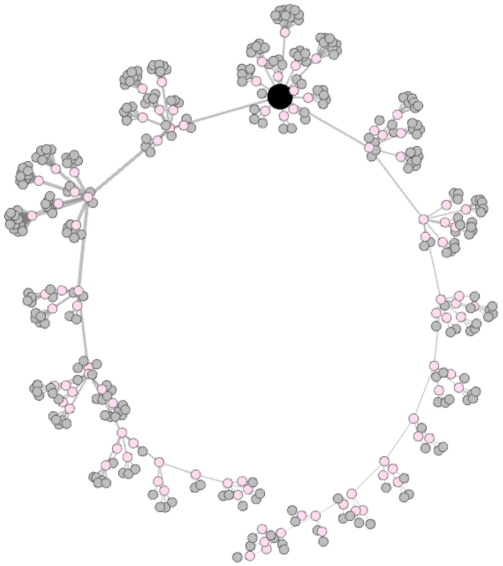
Function EdgeWeightCalculation()
Input: network topology, set of sensors and set of data types
Output: complex weighted networks for each data type

1: **for all** $dty_x \in |DTY|$ **do**
2: totalWeightedGraph = 0
3: **for all** $sns_w \in dty_x$ **do**
4: snsWeightedGraph = calculateEdgeWeight(sns_w)
5: totalWeightedGraph += snsWeightedGraph
6: **end for**
7: setCN[dty_x] = totalWeightedGraph
8: **end for**
9: **return** setCN

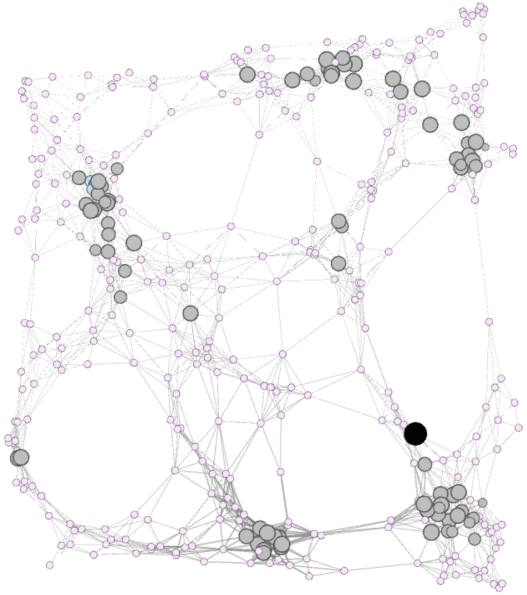
Experimental evaluation

- Four network topologies:
  - Barabasi-Albert, Random Euclidean, Lobster, Grid
- Three centrality indices:
  - Eingenvector, Betweenness, Current flow betweenness
- Four experiment sizes:
  - Number of data types: 50, 100, 200, 300
- Simulator: YAFS (https://github.com/acsicuib/YAFS)

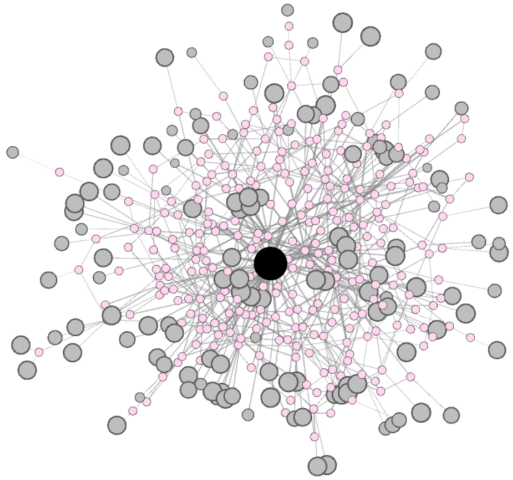
Experimental evaluation



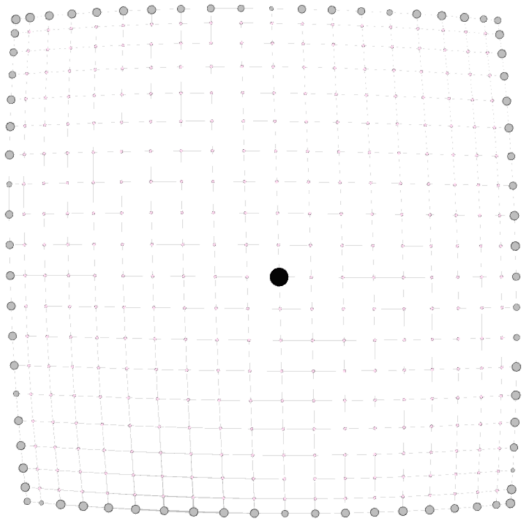
(a) Lobster topology



(b) Random Euclidean topology



(c) Barabasi-Albert topology



(d) Grid topology

Device capacity = 1.0

DT size = 1.0

Data rate = [1/10..1/50]

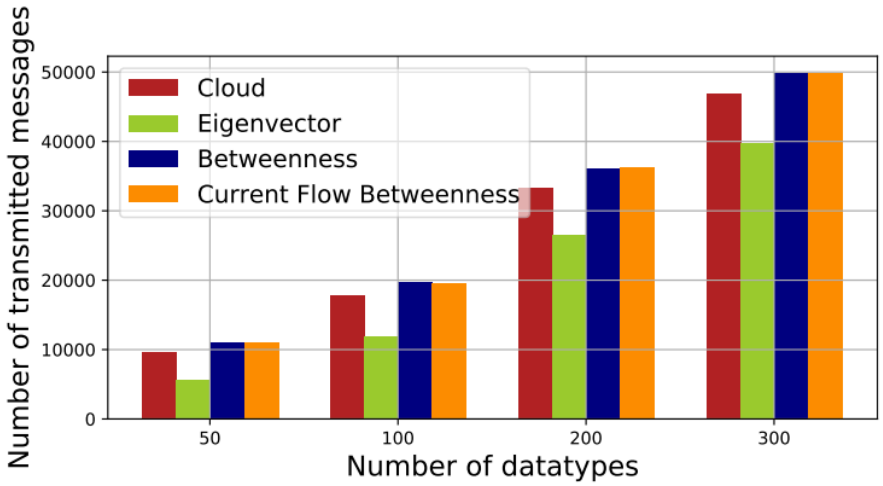
Proximity-based dist. of the sensors

Link latency = 1.0

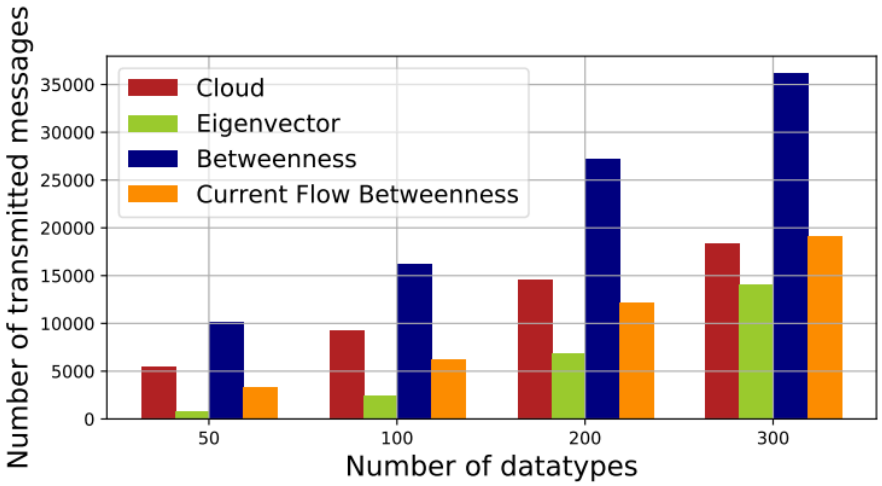
Packet size = 500 bytes

400 devices

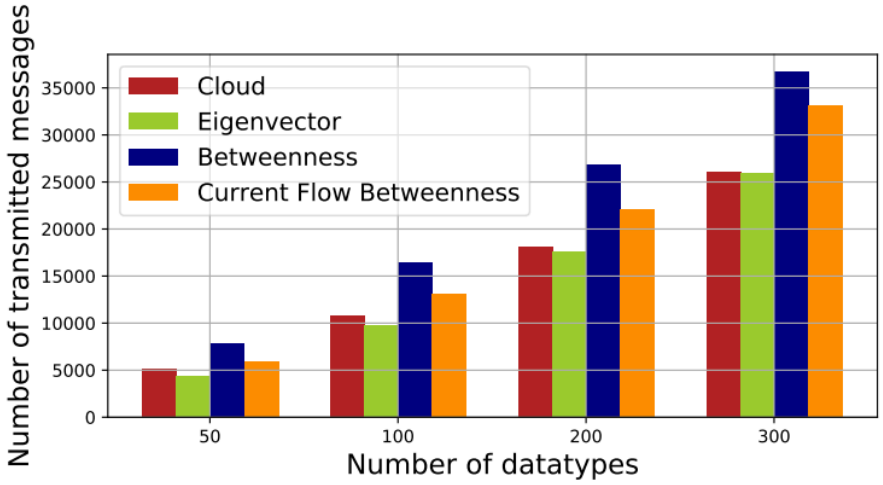
Results



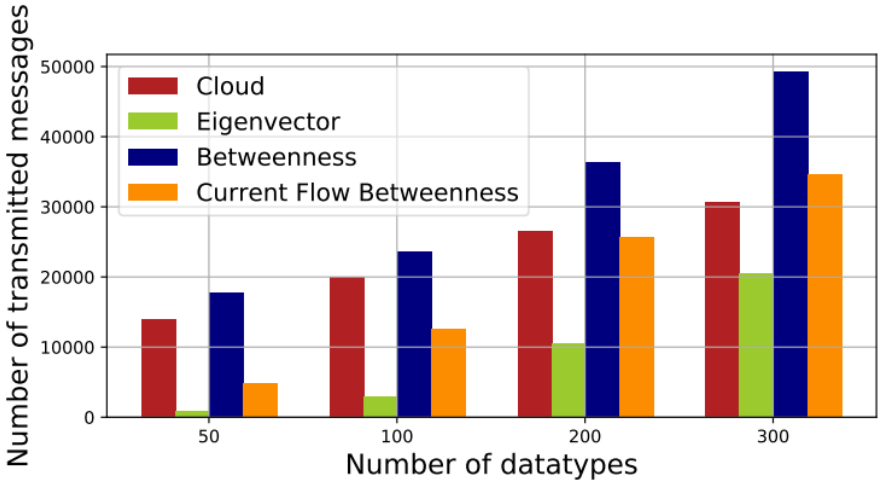
(a) Lobster topology



(b) Random Euclidean topology



(c) Barabasi-Albert topology



(d) Grid topology

Discussion

- Best centrality index: Eigenvector
- Worst centrality index: Betweenness

- Best topologies: Grid and Random Euclidean

- Increase of data types → Decrease of our approach improvements
  - Increase of data conflicts in the data nodes

Conclusion

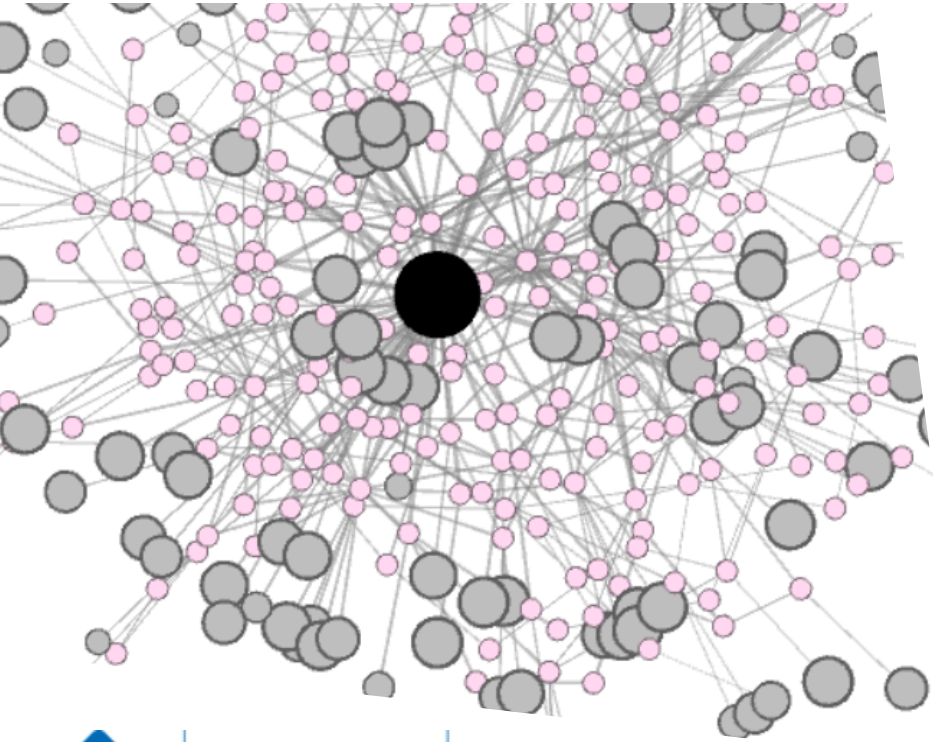
- Data placement approach for fog architecture
  - Store the data as closer as possible to the data generators
  - Based on centrality indices
  - Improvement of number of packet transmissions

- Future works
  - Relation between topology features and results
  - Characterization of real fog architectures

  - Include a more sophisticated storage domain
    - Replicas, chunks, …

Comparing Centrality Indices for Network Usage Optimization of Data Placement Policies in Fog Devices

Isaac Lera, Carlos Guerrero, Carlos Juiz



Thank you!!!!
Q & A

carlos.guerrero@uib.es
@guerrerotome
http://www.guerrerotome.com

This work was supported by the Spanish MINECO and the European Commission (FEDER funds) throught the grant number TIN2017-88547-P

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA, INDUSTRIA Y COMPETITIVIDAD
AGENCIA ESTATAL DE INVESTIGACIÓN
FEDER
Fondo Europeo de Desarrollo Regional
UNIÓN EUROPEA
"Una manera de hacer Europa"

UIB Universitat de les Illes Balears
#SOM UIB

April 23rd, 2018
FMEC 2018, Barcelona