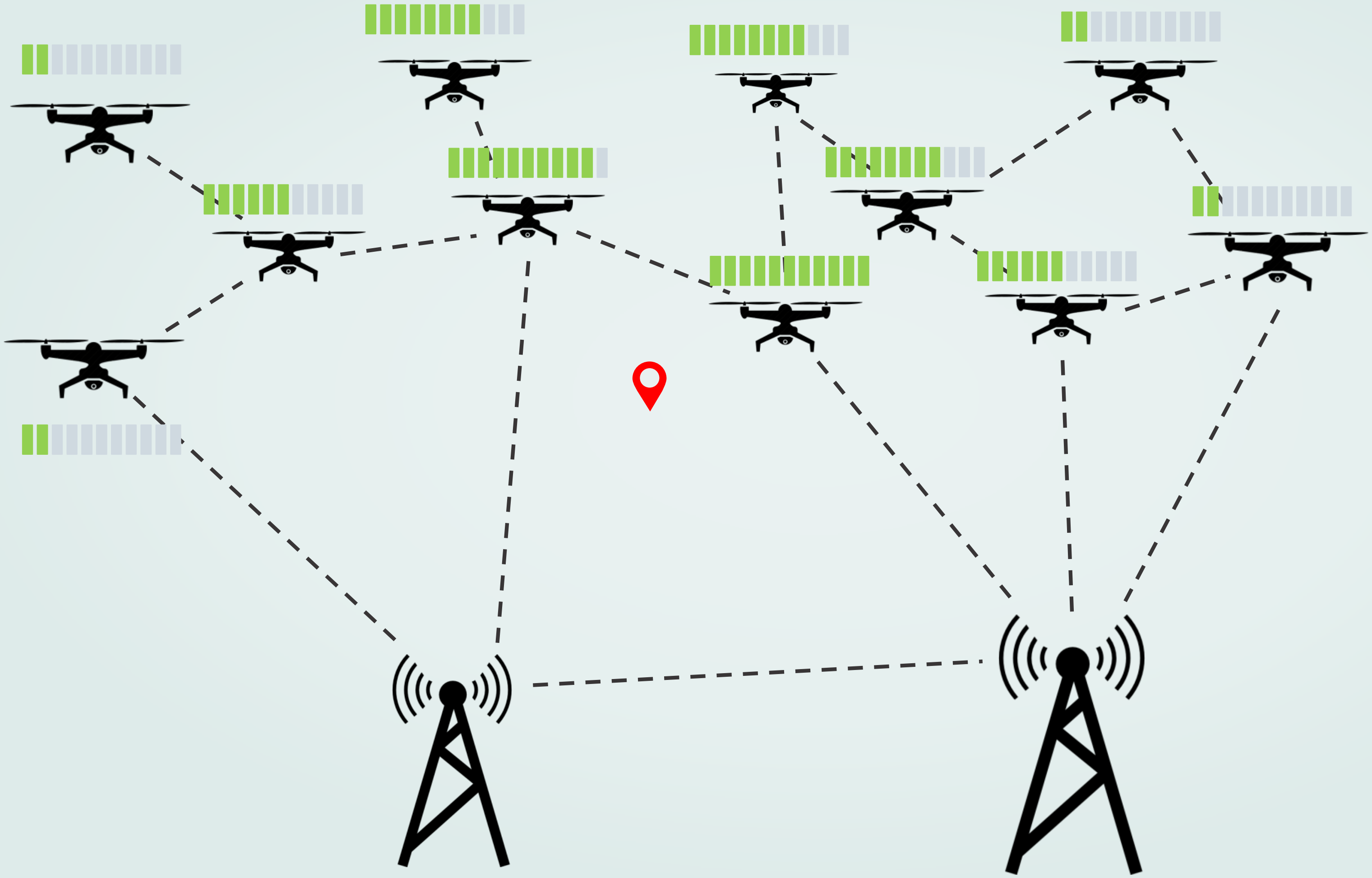
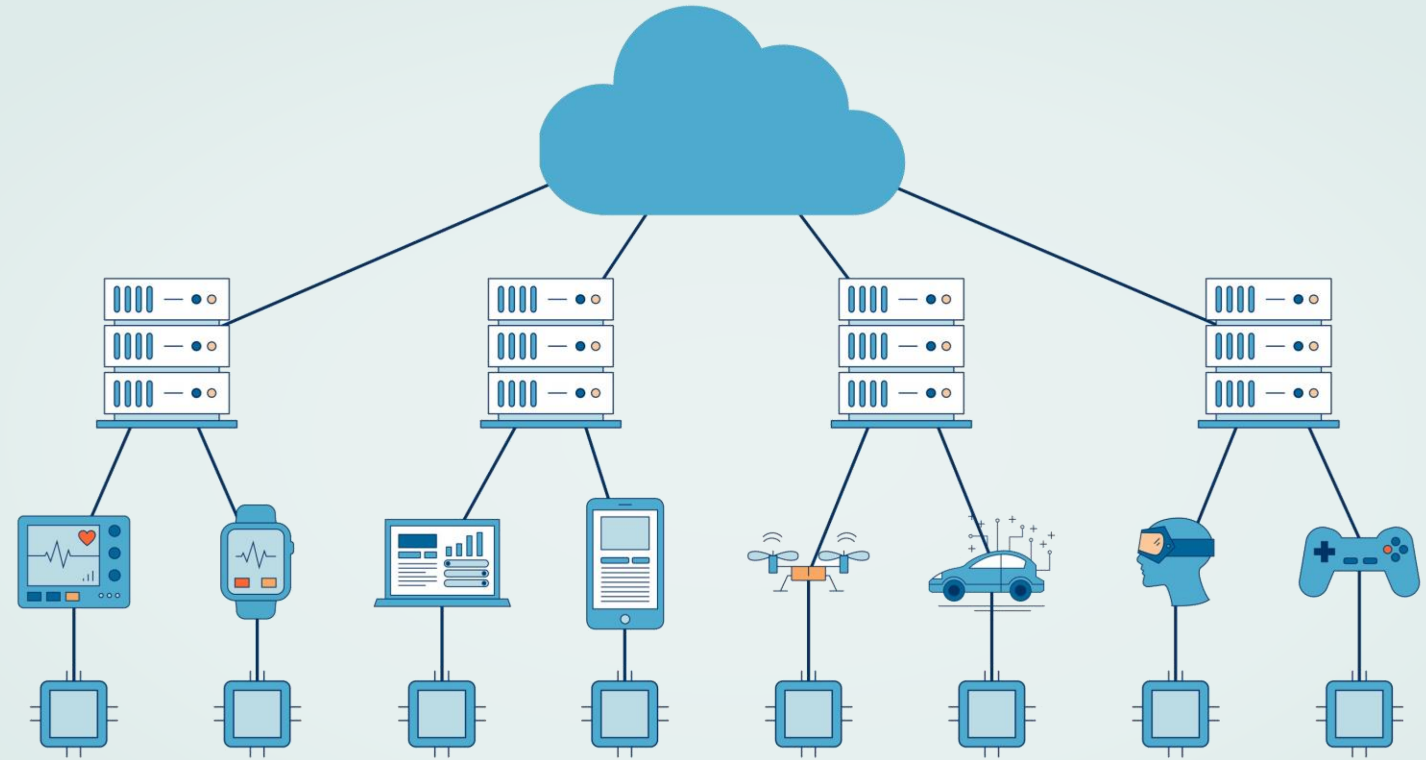
Modeling Practical Distributed Systems

Hasan Heydari
ReSCo, ENAC, Université de Toulouse





Problem

Consider a non-uniform distributed system. How can we design algorithms for such a system, where each process for collaborating with other processes should consider its weight and the weights of others?

The roadmap

01

Weight Assignment

Some weight(s) should be assigned to each process

02

Classification

Based on the assigned weights can we classify non-uniform distributed systems?

03

Computing Model

A computing model should be presented to determine how an algorithm designer can use the weights

04

Applications

Where and when can one use this model?

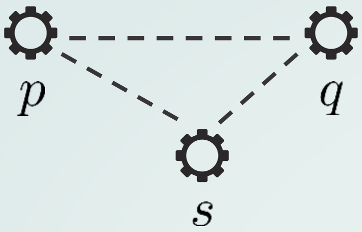
05

Future Works

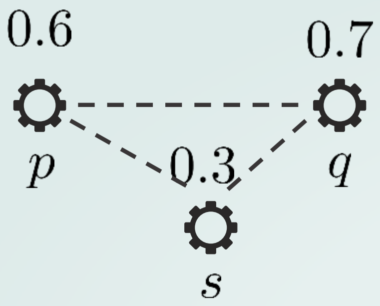
How can we contribute this work?

01

Weight Assignment



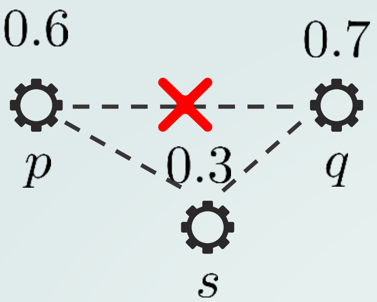
$$W(process, time)$$



$$W(process, time)$$

Important Question

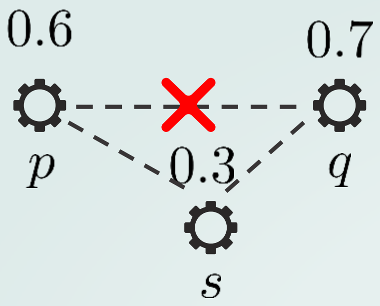
Does *W* satisfy all the requirements of our problem?



$$W(process, time)$$

Important Question

Does *W* satisfy all the requirements of our problem?



$$W(process, time)$$

Important Question

Does W satisfy all the requirements of our problem?

$$OW(observer, observed\ process, link, time)$$

Each process is an observer and an observed one.

02

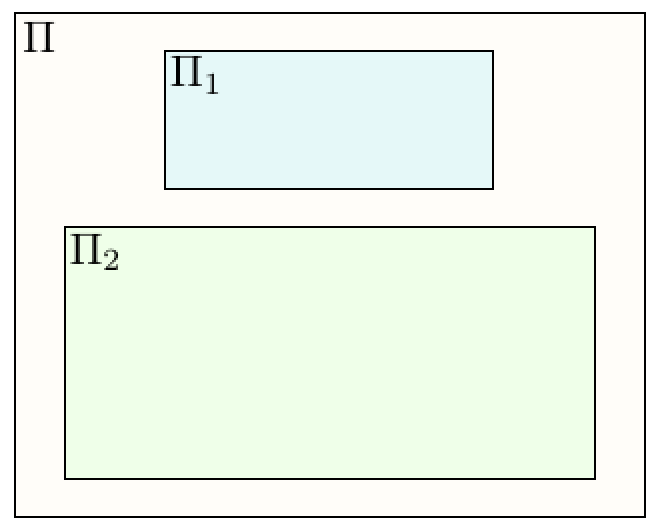
Classification

Class 1.

OW is independent of the observer p, the observed process q, the communication link l, and the time instance t, and is equal to a constant c

Class 3.

There are k time-invariant groups $\{g_1, g_2, \ldots, g_k\}$, where k is a constant. For every two groups g_i and g_j, an ordered pair (g_i, g_j) states that the processes of g_i are observers and observe the processes of g_2. Each pair has its own weight that is time-invariant. Indeed, if pair (g_i, g_j) has weight $c_{i,j}$, the processes of g_i observe the weights of processes of g_j with weight $c_{i,j}$.



Consider a distributed system consisting of $n{=}100$ nodes. Assume that there exist $n_1{=}10$ (Π_1) and $n_2{=}90$ (Π_2) nodes in the core and edge of the network respectively. Consider that:

$$w(q \rightsquigarrow p) = \begin{cases} 0.99 & \text{if } p, q \in \Pi_1 \\ 0.8 & \text{if } p, q \in \Pi_2 \\ 0.5 & \text{if } (p \in \Pi_1 \land q \in \Pi_2) \lor (p \in \Pi_2 \land q \in \Pi_1) \end{cases}$$

03

Computing Model

Heard-Of Model

Each process p has a local round number r_p.

Each process p, before sending a message, attaches r_p to the sending message ($m_p^{r_p}$), then sends $m_p^{r_p}$ to *all* processes, and any process takes into account just those received messages that have the same round numbers as its local round number.

Let $\mu_p^{r_p}$ denote the set of messages that p takes into account in round r_p and $HO(p, r_p)$ be the set of processes that sent the messages of $\mu_p^{r_p}$, i.e, $HO(p, r_p) = \{q : m_q^{r_q} \in \mu_p^{r_p}\}$.

The general scheme of designed algorithms in the HO model is to check the size of the set $HO(p, r_p)$ for each process p, and if its size is more than a defined threshold, some computation is done based on the content of $\mu_p^{r_p}$.

Since the concept of observation (observer and observed processes) is similar to the concept of Heard-Of, we choose this model to generalize it for considering non-uniformity and modeling non-uniform distributed systems.

Non-Uniform Heard-Of (NUHO) Model

The general scheme of designed algorithms in the Non-Uniform HO model is to check the weight of the set $NUHO(p, r_p)$ for each process p, and if its weight is more than a defined threshold, some computation is done based on the content of $\mu_p^{r_p}$.

04

Applications

Improving the performance of leader-less consensus algorithms

Improving the performance of leader-less state machine replications

05

Future Works

Adapting the model for the case that the number of the nodes or links are changing …

Finding other problems to solve with this model

Hasan Heydari (heydari@enac.fr)

ENAC, Université de Toulouse