Algoritmos e Sistemas Distribuídos: Cyclon Algorithm

João Carlos Antunes Leitão
NOVA Laboratory for Computer Science and Informatics (NOVA LINCS)
and
Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade NOVA de Lisboa

2019

Algorithm 1: Cyclon (Unstructured Overlay Management)

Interface:
Requests:
getNeighbors ( )
Indications:
neighbors (n **)** `//n is the set of neighbors of the local process`

State:
neigh `//set of neighbors of the process (local partial view)`
N `//maximum number of neighbors`
sample `//Sample of neigh sent to the other process in last shuffle`

Upon Init ($contact$**,** $maxN$**) do:**
N $\longleftarrow$ $maxN$;
If contact $\neq \perp$ **then**
neigh $\longleftarrow$ $\{(contact, 0)\}$;
else
neigh $\longleftarrow$ $\{\}$;
sample $\longleftarrow$ $\perp$;
Setup Periodic Timer Shuffle (T); `//T is the shuffle period, in the order of seconds`

Upon getNeighbors( ) do:
pview $\longleftarrow$ neigh; `//To avoid the upper layer to modify the neigh set`
Trigger neighbors(pview);

Upon Shuffle( ) do:
foreach $(p, age) \in$ neigh **do**
neigh $\longleftarrow$ (neigh $\setminus (p, age)) \cup (p, age + 1)$;
p $\longleftarrow$ pickOldest(neigh);
If p $\neq \perp$ **then**
neigh $\longleftarrow$ neigh $\setminus$ p;
sample $\longleftarrow$ randomSubset(neigh);
Trigger Send (ShuffleRequest, p**,** $sample \cup \{(myself, 0)\}$**)**;

Upon Receive (ShuffleRequest, s**,** $peerSample$**) do:**
temporarySample $\longleftarrow$ randomSubset(neigh);
Trigger Send (ShuffleReply, s**,** $temporarySample$**)**;
Call mergeViews($peerSample$**,** $temporarySample$**)**;

Upon Receive (ShuffleReply, s**,** $peerSample$**) do:**
Call mergeViews($peerSample$**,** $sample$**)**;

Procedure mergeViews($peerSample$**,** $mySample$**)**
Foreach $(p, age) \in$ peerSample **do**
If $(p', age') \in$ neigh $\wedge$ $p' = p \wedge age' > age$ **then**
neigh $\longleftarrow$ (neigh $\setminus (p', age')) \cup \{(p, age)\}$;
Else If $\#$neigh $< N$ **then**
neigh $\longleftarrow$ neigh $\cup \{(p, age)\}$;
Else
(x, age') $\longleftarrow$ (x,age') : (x,age') $\in$ neigh $\wedge$ (x,age") $\in$ $mySample$; `//Pick an element of neigh that is also in mySample`
If (x, age') $= \perp$ **then**
(x, age') $\longleftarrow$ (x,age') : (x, age') $\in$ neigh; `//Pick a random element of neigh`
neigh $\longleftarrow$ (neigh $\setminus (x, age')) \cup \{(p, age)\}$;