1

Jeder Knoten v führt folgenden Algorithmus aus:

Runde 1

Ist $v \in L$:

Sende "LEADER"-Nachricht an Nachbarn im Uhrzeigersinn.

Runde 2 ... n

Wird "LEADER"-Nachricht empfangen:

Ist $v \in L$:

Sende "ERROR"-Nachricht im Uhrzeigersinn

Ist $v \notin L$:

Sende "LEADER"-Nachricht im Uhrzeigersinn

Wird "ERROR"-Nachricht empfangen:

L ist ungültig

Ist $v \notin L$:

Sende "ERROR"-Nachricht im Uhrzeigersinn

Runde n+1 ... 2n−1

Wird "ERROR"-Nachricht empfangen:

L ist ungültig

Ist $v \notin L$:

Sende "ERROR"-Nachricht im Uhrzeigersinn

Runde 2n

Ist L noch nicht ungültig, so ist L gültig.

Laufzeit

Der Algorithmus terminiert auf jeden Fall in $2n+1 = O(n)$ Runden.

Korrektheit, Nachrichtenkomplexität

Fall 1: $|L| = 1$

Sei $z \in L$

Sei v_i, $i \in \{1, \dots, n\}$ der Knoten mit Distanz i im Uhrzeigersinn zu z.

Induktionshypothese:

In Runde $i+1$ empfängt v_i die Nachricht "LEADER".

Induktionsbasis: $i = 1$

z sendet "LEADER" in Runde 1 an v_1. v_1 empfängt diese Nachricht in Runde 2.

Induktionsschritt: $i \to i+1$

v_i empfängt "LEADER" in Runde $i+1$. $v_i \notin L$, also sendet v_i "LEADER" in Runde $i+1$ an v_{i+1}. v_{i+1} empfängt diese Nachricht in Runde $i+2$.

Nach $n+1$ Runden haben alle Knoten 1x "LEADER" empfangen. Es wurden insgesamt n Nachrichten versendet. Entsprechend können keine "ERROR"-Nachrichten versandt worden sein.

In Runde $n+1$ empfängt $v_n = z$ "LEADER". Da wir in Runde $n+1$ sind, geschieht nichts.

In Runde $2n+1$ wissen alle Knoten, dass L gültig ist.

Insgesamt wurden n Nachrichten versendet.

Fall 2: $|L| \geq 2$

Sei $z_i \in L$, sei $z_{i+1} \in L$ der im Uhrzeigersinn nächste Knoten in L zu z_i. Sei weiters z_{i+2} analog definiert.

Analog zu obigem Induktionsbeweis lässt sich zeigen, dass z_{i+1} in Runde $dist(z_i, z_{i+1}) + 1$ die Nachricht "LEADER" empfangen wird.

Da $z_{i+1} \in L$, wird z_{i+1} "ERROR" weitersenden. Wieder analog zu obiger Induktion wird z_{i+2} diese Nachricht in weiteren $dist(z_{i+1}, z_{i+2})$ Runden empfangen.

Jeder Knoten v aus dem Ring liegt zwischen zwei Leadern.

Es gilt also, dass nach Runde

$$\max_{z_i}\{dist(z_i, z_{i+1}) + 1 + dist(z_{i+1}, z_{i+2})\}$$

jeder Knoten die Nachricht "ERROR" empfangen hat.

$\forall u, v$ gilt $dist(u,v) \leq n-1$

Damit gilt

$$\max_{z_i}\{dist(z_i, z_{i+1}) + 1 + dist(z_{i+1}, z_{i+2})\}$$
$$\leq \max_{z_i}\{n-1+1+n-1\}$$
$$= 2n-1$$

Damit weiß jeder Knoten vor der "Finalen" Runde $2n$, dass L ungültig ist.

Die Anzahl versandter Nachrichten ist

$$2\sum_{z_i \in L} dist(z_i, z_{i+1}) = 2n = O(n)$$

② Jeder Knoten v führt folgenden Algorithmus aus:

Runde 1:

Setze $minId = \infty$

Sende $ID(v)$ an alle Nachbarn

Runde 2...n:

Speichere das Minimum aller empfangenen IDs als $minId_{neu}$

Ist $minId_{neu} < minId$:

Setze $minId = minId_{neu}$

Sende $minId$ an alle Nachbarn

Runde $n+1$:

Ist $minId = ID(v)$:

Werde Leader

sonst:

Werde Follower

Laufzeit

Der Algorithmus terminiert auf jeden Fall in $n+1 = O(n)$ Runden.

Korrektheit

Sei z der Knoten mit der kleinsten ID.

Induktionshypothese:

In Runde $i+1$ gilt für alle Knoten v mit $dist(z,v) \leq i$: $v.minId = ID(z)$

Induktionsbasis:

In Runde 2 erhalten die Nachbarn von z (d.h. $dist(z,v) \leq 1$) die Nachricht $ID(z)$. Da z die kleinste ID hat, speichern alle $ID(z)$ als minId.

Induktionsschritt: $i \to i+1$

Die Hypothese gilt per starker Induktion für alle v mit $dist(z,v) < i+1$. z.Z. bleibt $dist(z,v) = i+1$. Sei also v sodass $dist(z,v) = i+1$.

Da $dist(z,v) = i+1$ muss $\exists v' : dist(z,v') = i \wedge dist(v',v) = 1$

Per IH ist $v'.minId = ID(z)$. v' hat die Nachricht $ID(z)$ in Runde i erhalten (da $dist(z,v') = i$). Entsprechend erhält v als Nachbar von v' die Nachricht $ID(z)$ in Runde $i+1$.
Da $ID(z)$ die kleinste ID ist, speichert sie v in minId.

Da $\max_v \{dist(z,v)\} \leq n$ (CONGEST ist zusammenhängend), gilt in Runde $n+1$ $\forall v$: $v.minId = ID(z)$.

Dann wird z Leader und alle anderen werden Follower.

3

Vorgangsweise analog zum randomisierten Leader Election Algorithmus aus der VO.

- Es gibt einen Las-Vegas-Algorithmus für P, der erwartete Laufzeit $R(n)$ hat

- Sei q die Wahrscheinlichkeit, dass der LV-Algo nach $R(n)$ Runden terminiert

- Sei $k = c \cdot \log_{(1-q)^{-1}} n$

- Die Wahrscheinlichkeit, dass der LV-Algo nach $k \cdot R(n)$ Runden noch nicht terminiert ist, ist:

$$(1-q)^k = (1-q)^{c \log_{(1-q)^{-1}} n}$$
$$= ((1-q)^{\log_{(1-q)^{-1}} n})^c$$
$$= (1/n)^c$$
$$= 1/n^c$$

- Die Wahrscheinlichkeit, dass der LV-Algo nach $k \cdot R(n)$ Runden bereits terminiert ist, ist demnach:

$$1 - 1/n^c$$

- Die Laufzeit ist:

$$O(k \cdot R(n)) = O((c \cdot \log_{(1-q)^{-1}} n) \cdot R(n)) = O(R(n) \cdot c \log n)$$