EDA387: Computer Networks - Lab 2.1

Mats Högberg & Juliana Siburian

September 26, 2019

Value discovery in complete graphs

With processor identifiers

An algorithm for letting each processor p_i discover its secret value s_i for any value of n is:

```
01: Do forever:
02:    r_i := read s_{(i-1) mod n}
03:    secret := read r_{(i+1) mod n}
04: End
```

Assume that the system is asynchronous, that all processors execute the same program, and that the values in the s-registers never change. Let $E = (c_0, c_1, \ldots)$ be a system execution of the above algorithm, where c_0 is an arbitrary starting configuration and c_r is the configuration after round r.

After at most 2 rounds all processors will have executed line 02. On this line, r_i is set to $s_{(i-1) \mod n}$. Since this is the only line where r_i is updated, and s_i never changes for any processor, we have that $\forall i : r_i = s_{(i-1) \mod n} \Leftrightarrow \forall i : r_{(i+1) \mod n} = s_i$ for all configurations c_r where $r \geq 2$.

After at most 4 rounds all processors will also have executed line 03, in which the value of the `secret` variable is set to $r_{(i+1) \mod n}$. This is the only line where the `secret` variable is set, and we already proved that $\forall i : r_{(i+1) \mod n} = s_i$ for all $c_r, r \geq 2$, which means that `secret` $= s_i$ for all processors p_i for all configurations c_r where $r \geq 4$.

Thus, we have proved both convergence and closure for the algorithm.

Without processor identifiers

Let $s_{\max}$ be the size of the s-registers and $r_{\max}$ be the size of the r-registers. Assuming that $r_{\max} \geq (n-1)s_{\max}$, we can use the following algorithm for letting each processor p_i discover its secret value s_i in a network without processor identifiers:

```
01: Do forever:
02:    r_i := sum { read s_j for all other processors p_j }
03:    secret := r_k + s_k - r_i, where p_k is any other processor
04: End
```

The proof for this algorithm is very similar to the proof for the previous algorithm. We assume that the system is asynchronous, that all processors execute the same program, and that the values in the secret registers never change. Let $E = (c_0, c_1, \ldots)$ be a system execution of the algorithm, where c_0 is an arbitrary starting configuration and c_r is the configuration after round r.

After at most $n+1$ rounds all processors will have executed line 02. On this line, the values in the s-registers of all other processors are read, and the sum of them are stored in r_i. We thus have that $r_i = \sum_j s_j - s_i$. Since this is the only line where r_i is written and since the values in the s-registers never change, the values of the r-registers will never change after this round. Thus, we have that $\forall i : r_i = \sum_j s_j - s_i$ holds for all configurations c_r where $r \geq n + 1$.

After at most $n + 1$ more rounds all processors will also have executed line 03. On this line, the value of the `secret` variable is set to $r_k + s_k - r_i$, where p_k is any other processor. We have already proved that $\forall i : r_i = \sum_j s_j - s_i$ holds for every round $c_r, r \geq n + 1$, so the value of the `secret` variable for an arbitrary processor p_i after round $2n + 2$ can be written as:

$$\text{secret} = r_k + s_k - r_i = \sum_j s_j - s_k + s_k - \Big(\sum_j s_j - s_i \Big) = s_i$$

And since this is the only line where `secret` is updated, we have that `secret` $= s_i$ for all processors p_i for all configurations c_r where $r \geq 2n + 2$.

Thus, we have proved both convergence and closure for the algorithm.

Note that in the problem statement it says that the sizes of the s and r-registers are constant and independent of n, but here we assume that $r_{\max} \geq (n-1)s_{\max}$. We haven't been able to come up with a solution for when this does not hold, but we haven't been able to prove that no such solution exists either.