Home assignment for Computer Network EDA387

Group 24

September 18, 2025

Problem Consider a set $P = \{p_0, \ldots, p_{n-1}\}$ of n processors, such that each processor p_i is associated with two registers r_i and s_i (both of constant size independent of n). Processor p_i cannot read the value in s_i, however, any other processor can. Processor p_i has both read and write access rights to r_i and any other processor has only read-only access rights to r_i. The value in s_i is unknown to p_i. The other processors can help p_i discover this unknown value. For example, suppose that $n = 2$. Processor $p_{(i+1) \bmod 2}$ can write the value of s_i to $r_{(i+1) \bmod 2}$ and then p_i can discover s_i's value by reading $r_{(i+1) \bmod 2}$. Please solve the problem, which is to let p_i discover the secret s_i, for the case in which $n > 2$ is any finite known value.

Question 1

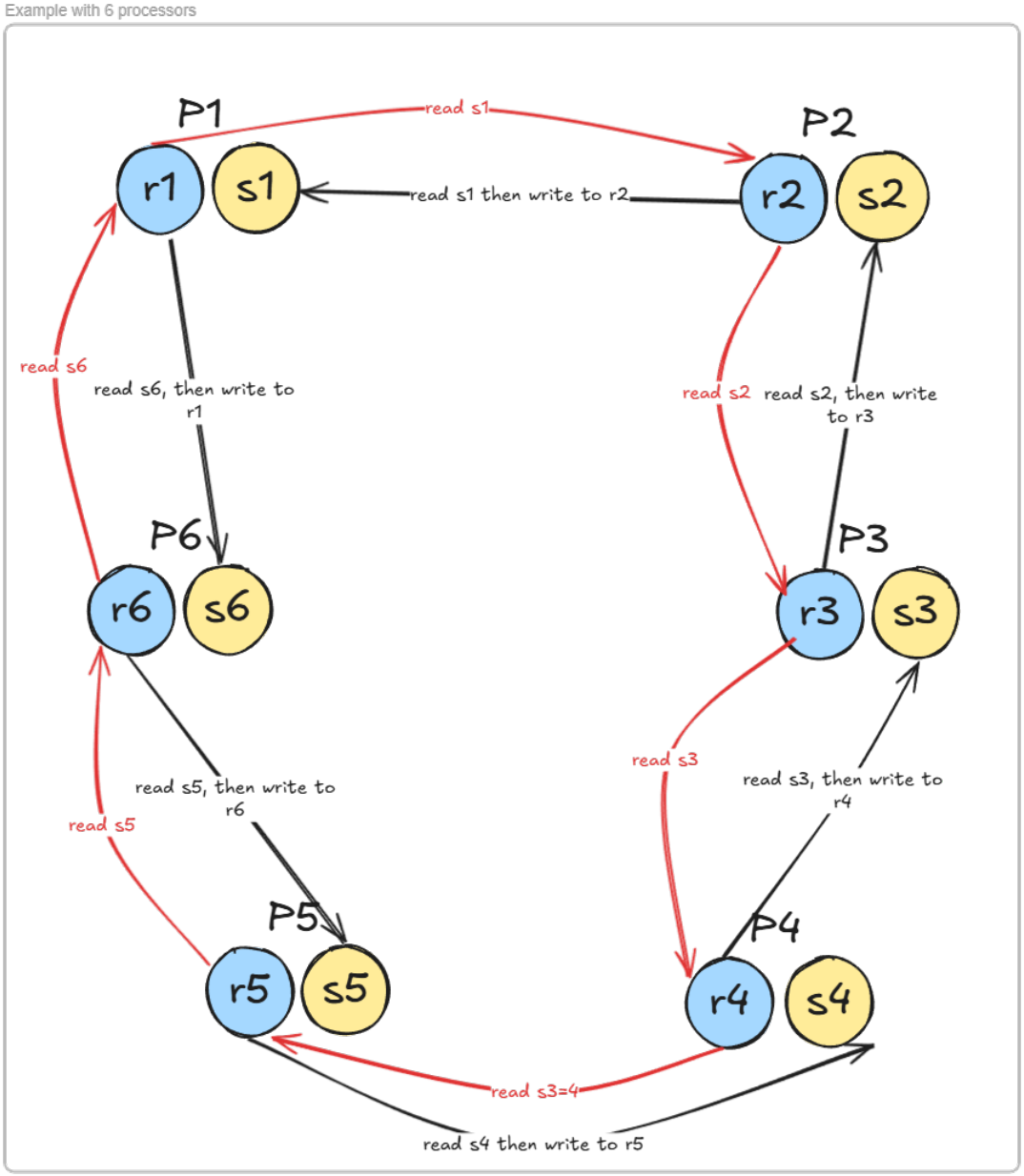
Assume that all processors have access to a globally unique identifier within the range $\{0, \ldots, n-1\}$, i.e., the identifier is defined by a one-to-one function $\mathrm{ID} : P \to \{0, \ldots, n-1\}$. For instance, $\forall p_i \in P : \mathrm{ID}(p_i) = i$, i.e., the identifier of processor p_i is equal to its index i. In practice, MAC addresses are generally assumed to be globally unique. Is there a solution when processors have globally unique identifiers? Prove your claims.

Proof. The answer for this question is *cyclical*. Each processor can write the secret of its next processor and so on, until the last one which writes the secret from the first processor. Let the processor p_i be the selected one. It has two registers: r_i and s_i. The process to get the value of s_i can be divided into two stages: **Discovery** and **Read**.

- **Discovery.** Processor p_i can read the value in the register $s_{(i-1) \bmod n}$ and write it to r_i.
- **Read.** When the discovery phase is completed, processor p_i can read the value in $r_{(i+1) \bmod n}$. By construction, $r_{(i+1) \bmod n}$ contains s_i, so p_i learns its own secret.

As stated before, only p_i has the right to modify the content inside register r_i, after the discovery step, inside r_i of any p_i, there will be the value of $s_{(i-1) \bmod n}$. Therefore, any processor p_i can read the content inside the register $r_{(i=1) \bmod n}$ to obtain the value of s_i. For each processor, only itself has write-privilege to r_i. Moreover, other processors can only read r_i and s_i, hence, the content can only be modified by p_i. This guarantees the content of s_i that p_i discovered through neighbor is correct. With this approach, each processor has to do two operations (discover - reading from others and reading - obtaining its own secret from others) with complexity of O(n) each.

Example with 6 processors



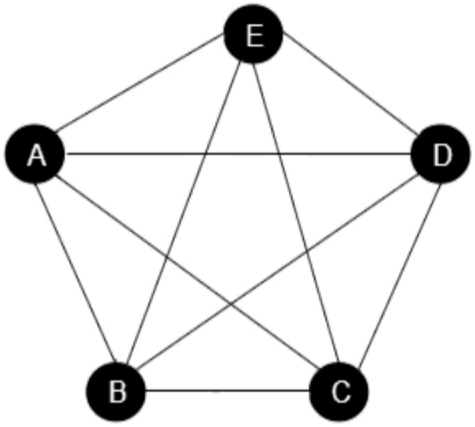
□

Question 2

Now, suppose processors only have access to *locally unique identifiers*. Specifically, let $N(p_i) \subseteq P \setminus \{p_i\}$ denote the neighborhood of processor $p_i \in P$, which is the set of all processors directly connected to p_i. For each $p_i \in P$, the local identifier is defined by a one-to-one function $\mathrm{ID}_i : P \to \{0, \ldots, n-1\}$. Note that $\mathrm{ID}_i()$ depends on p_i, whereas ID() is identical for all processors.
Furthermore, for processors $p_i, p_j, p_k \in P$, it may be the case that $\mathrm{ID}_i(k) \neq \mathrm{ID}_j(k)$. For example, port numbers are unique to the host but not across the Internet.
Is there a solution when processors have only locally unique identifiers? Prove your claims.

Proof. Considered the selected processor p_i
Since the communication graph is complete, every processor is connected to every other processor (i.e., each processor is a neighbor of all others). Processor p_i also has its neighbor set.

- Each processor will write down each other's secrets as in the **Question 1**. When this process is done, any processor can inspect its own r_i register to know that every secrets are present, *except one*. That missing information must therefore be its own s_i.
- More generally, processors can use a distributed algorithm to discover the network topology. With these information, each processor p_i must find a path to the processor that can write its secret s_i to some register that p_i itself can read.
- In this scenario, only *local identifiers* are available, processors must determine the network structure first, this, use this information to coordinate the sharing of secrets
- With the neighbor set for each processor and the exchange of local identifiers, processor can learn the complete network topology.



□