1 Introduction

Today, Dr. Soma continued on Dissemination Quorum System. Recall the following theorem:

**Theorem 1** Let $BQS$ be a coterie over set $S$ and let $B$ be an adversary for $S$. Then, if there exists an asynchronous implementation of the SRSW storage on $BQS$, then $BQS$ is a dissemination quorum system.

We started the class by proving this theorem.

2 Dissemination Quorum System (Con’t)

**Proof:** Suppose, for contradiction, there is a SRSW safe storage emulation based on $BQS$ such that $BQS$ is not a dissemination quorum system. Therefore, $\exists Q_1, Q_2 \in BQS, \exists B \in B, Q_1 \cap Q_2 \subseteq B$. Consider $Q_1 \neq Q_2$.

Since $BQS$ is a coterie

$$Q_1 \notin Q_2 \land Q_2 \notin Q_1$$

Therefore,

$$Q_1 - Q_2 \neq \emptyset$$
$$Q_2 - Q_1 \neq \emptyset$$

Let

$$p_1 \in Q_1 - Q_2$$
$$p_2 \in Q_2 - Q_1$$

Let $p_1$ be a writer and $p_2$ be a reader and let:

$$B'_2 = B - \{p_2\}$$

Since

$$Q_1 \cap Q_2 \subseteq B \land p_2 \notin Q_1$$

We have

$$Q_1 \cap Q_2 \subseteq B'_2$$

1
Figure 1: This represents the situation where a node $p_m$ send Token message to node $j$, followed by request message.

We fix $B_2 \subseteq B'_2$ such that $Q_1 \cap Q_2 = B_2$

Since $B_2 \cap B'_2 \subseteq B$

We have $B_2 \subseteq B$

Consider the Figure 1, let $ex_1$ be partial execution in which all processors crash at the beginning except processors in $Q_1$.

In $ex_1$, $p_1$ invokes $write(v)$. Since there is a quorum that contains only correct processors ($Q_1$), $write(v)$ eventually completes in $ex_1$.

In $ex_2$, it is a partial execution where all processors crash at the beginning except processors in $Q_2$.

In $ex_2$, $p_2$ invokes $read()$. Since there is a quorum that contains correct processors ($Q_2$), read eventually happens. By definition of a safe register, read returns ⊥.

In partial execution $ex_3$, processors in $B_2$ are Byzantine. All remaining processors in $Q_1 \cap Q_2$ are correct. Messages sent by processors from $Q_1 \setminus Q_2$ are not delivered to $Q_2 \setminus Q_1$ (until later).

In $ex_3$,

- $p_1$ invokes $write(v)$. 

---

2
• $B_2$ follows protocol during $\text{write}(v)$ and $\text{write}(v)$ completes at time $t$. $ex_1 \sim_{Q_1-Q_2} ex_3$

• $B_2$ reverts to initial state at $t' > t$ and behaves as if they receive no message from $Q_1 - Q_2$.

At time $t'' > t'$, $p_2$ invokes $\text{read} \sim_{Q_2-Q_1} ex_3$. Since $p_2$ cannot distinguish $ex_3$ from $ex_2$, i.e., $ex_2 \sim ex_3$, read completes returning ⊥. This violates safeness property. Therefore, $BQS$ is a dissemination quorum system.

Definition 1 Given adversary $B = \{B||B| < t\}$, a dissemination quorum system is a $t$-dissemination quorum system if its resilience is at least $t$.

Lemma 2 In a $t$-dissemination quorum system $tDQS$, all quorums intersect in at least $t + 1$ elements.

Proof: For $\forall Q_1, Q_2 \in DQS$, and any $B \subseteq S$ where $|B| = t$, $Q_1 \cap Q_2 \not\subseteq B$. So, $|Q_1 \cap Q_2| > t$.

Lemma 3 No $t$-dissemination quorum system can be constructed over $S$ if $|S| \leq 3t$.

Proof: Assume, for contradiction, that there is a $t$-dissemination quorum system $tDQS$ over a set of processors where $n \leq 3t$. Then, by definition on $t$-resilient, there must be at least 2 quorums. Let $Q_1, Q_2 \in tDQS$ such that $|Q_1| \leq n - t$ and $|Q_2| \leq n - t$ and $|Q_2 - Q_1| = t$. Therefore, $|Q_2 \cap Q_1| \leq n - 2t \leq t$. So, $|Q_1 \cap Q_2| \leq t$. This is contradiction.