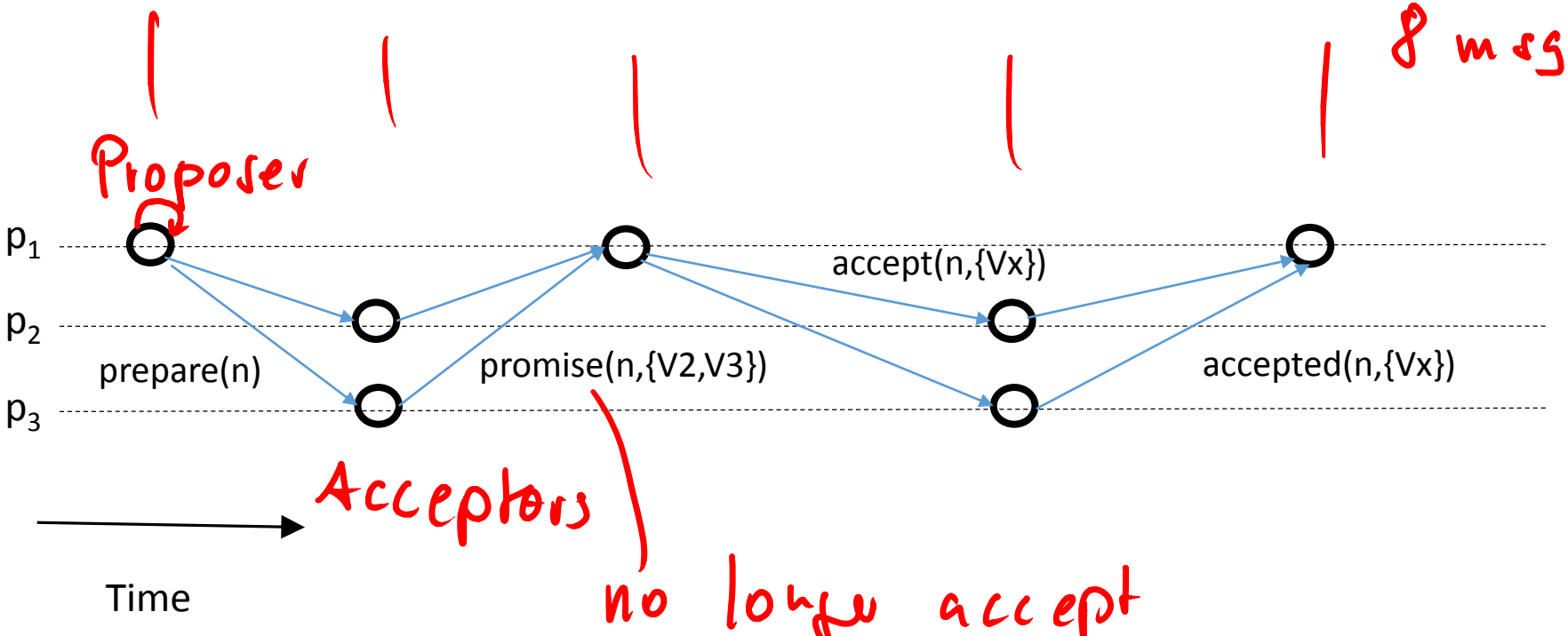
Distributed Systems

Exercise 5

a) p1 wants to publish a value and no failures occur.

4 rounds
8 msg



no longer accept proposals numbered < n

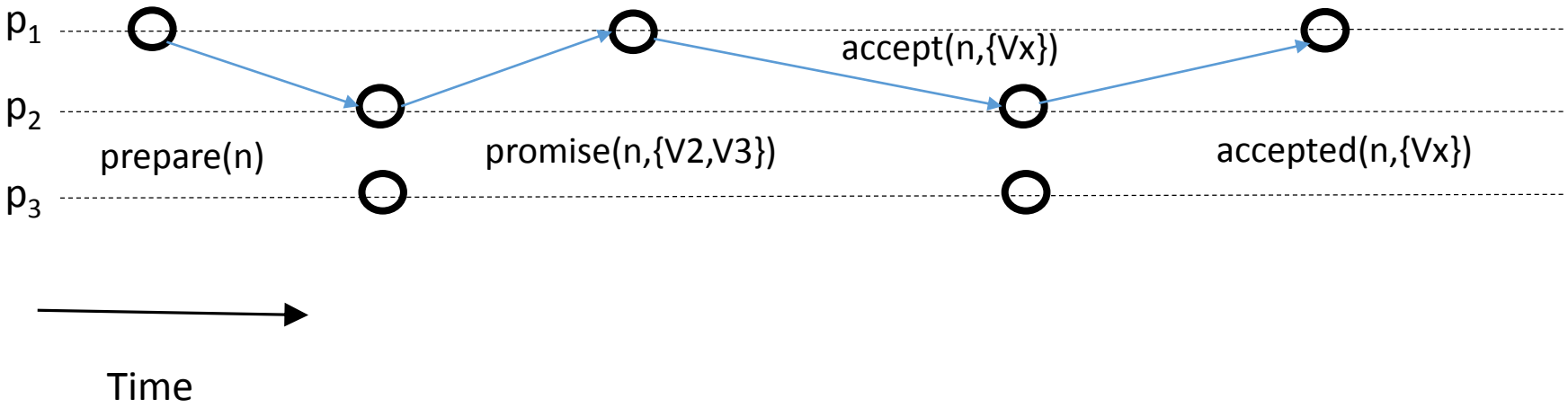
no promised values → Proposer chooses value

acceptors made promise before → need most recent value to propose / Prop chosen most recent

b) p1 uses and optimized version , where it only communicates with a quorum, again without failures!

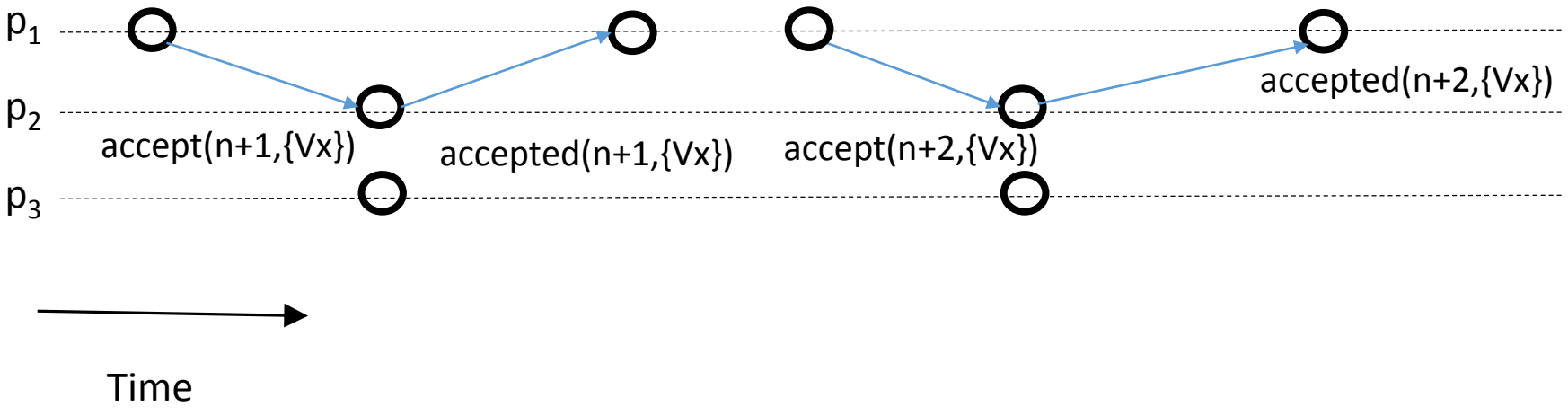
4 round trips

4 messages

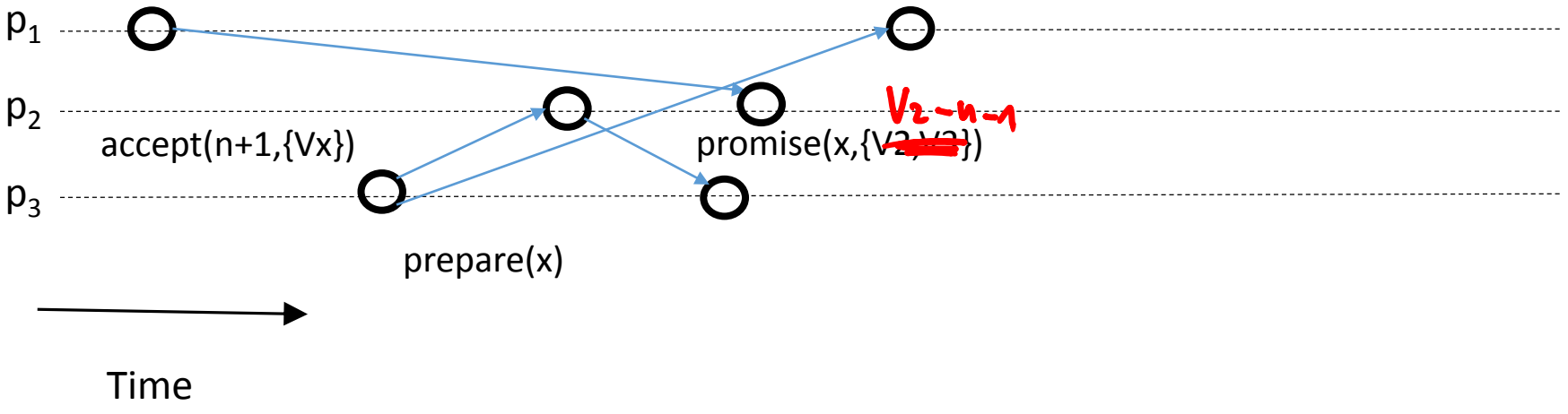


On failure of p2, start a new proposal with all acceptors → p3 → quorum reached!

c) Another iteration of b. optimized by using Multi-Paxos
(System is already in a steady state).



c) Another iteration of b. optimized by using Multi-Paxos
(System is already in a steady state).

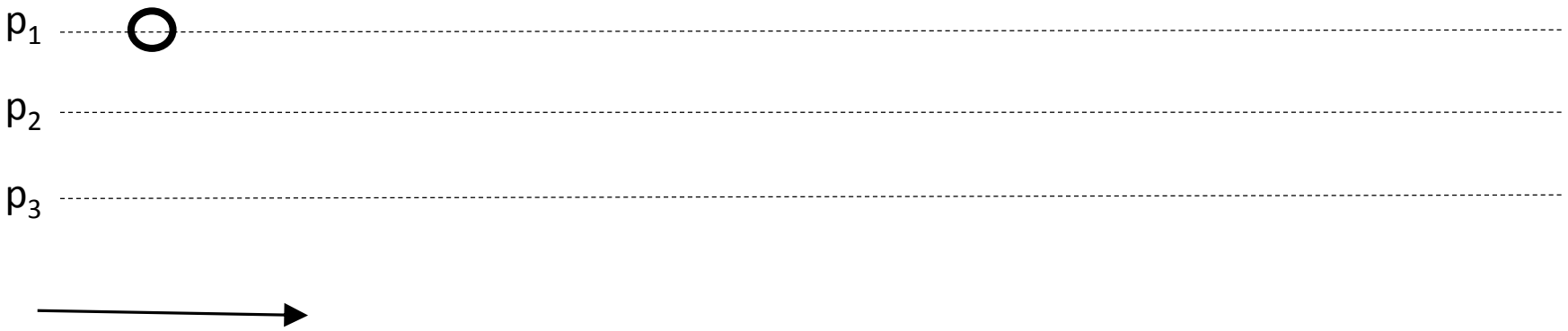


x smaller : deny (or ignore)

x greater : Pn promises + sends value

Pn fails acceptance => regular paxos

p_1 one dead in Multi-Paxos



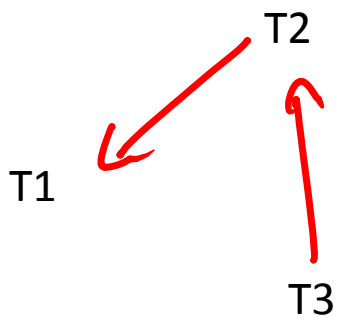
if $p_{2/3}$ - prepare(x) smaller than existing proposal number, keep proposing with increasing number

S_1: R_3X R_2Y W_2Y R_1Y W_1Y R_2X W_2X R_1X W_1X W_3Z.

X: $T_3 \rightarrow T_2 \rightarrow T_1$

Y: $T_2 \rightarrow T_1$

Z:



yes

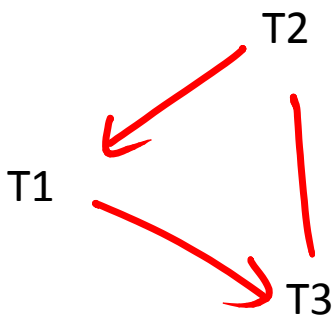
$T_3 \rightarrow T_2 \rightarrow T_1$

equivalent

S_2: R_3X R_2Y W_2Y R_1Y W_1Y R_2X W_2X R_1X W_1X W_3Y

X: $T_3 \rightarrow T_2 \rightarrow T_1$

Y $T_2 \rightarrow T_1 \rightarrow T_3$

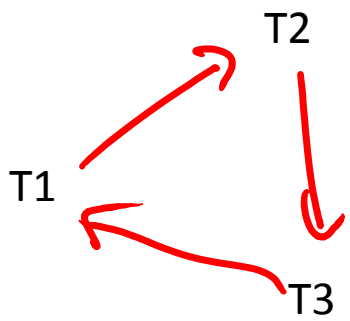


no → cycle

S_3: R_1Y W_1Y R_2Y W_2Y R_2X W_2X R_3Z W_3X R_1X W_1X.

X: $T_2 \rightarrow T_3 \rightarrow T_1$

Y: $T_1 \rightarrow T_2$



ห

T_1: RA WA
T_2: RA WA
T_3: RA WA

a) What is the number of serial schedules?

6 - all permutations

b) Which schedules exists that are **serializable**, but not **serial**?

0 same data access order

$R_i R_j \quad W_i W_j$

$R_i R_j \quad W_j W_i$

→ not possible due to conflicts

T1 : RA WC
T2 : RB WA
T3 : RC WD

a) How many schedules exist that are not serializable?

0 no cycle possible

b) Does using 2PL permit all serializable schedules?

S = R1A R2B W2A R3C W3D W1C

R_2B R_1A | W_2A W_1C R_3C W_3D

L_1A U_1A

not possible

$T_1 \rightarrow T_2$

$\searrow T_3$